L Number	Hits	Search Text	DB	Time stamp
59	9	(VoIP or "voice over IP") and ((transcoder near1 rate near1 adapt$5 near1 unit) or TRAU) with frame$1	USPAT; US-PGPUB; DERWENT	2004/03/17 11:18
63	1	(((transcoder near1 rate near1 adapt$5 near1 unit) or TRAU) with frame$1) and (admission adj control)	USPAT; US-PGPUB; DERWENT	2004/03/17 11:19
62	125	((transcoder near1 rate near1 adapt$5 near1 unit) or TRAU) with frame$1	USPAT; US-PGPUB; DERWENT	2004/03/17 11:21
66	16	(((transcoder near1 rate near1 adapt$5 near1 unit) or TRAU) with capacity)	USPAT; US-PGPUB; DERWENT	2004/03/17 12:05
73	7	(((transcoder near1 rate near1 adapt$5 near1 unit) or TRAU) with capacity) and ((voice or speech) and (packet or internet))	USPAT; US-PGPUB; DERWENT	2004/03/17 12:05
-	10	(VoIP or "voice over IP") and ("load management" or (admission near4 control)) and block$3 and capacity and (maximum with calls)	USPAT; US-PGPUB; DERWENT	2004/03/16 16:11
-	2	(VoIP or "voice over IP") and ("load management" or (admission near4 control)) and (TRAU with frame$1)	USPAT; US-PGPUB; DERWENT	2004/03/16 09:46
-	191	(VoIP or "voice over IP") and ("load management" or (admission near4 control))	USPAT; US-PGPUB; DERWENT	2004/03/16 10:45
-	8	(VoIP or "voice over IP") and ("load management" or (admission near4 control)) and "call blocking"	USPAT; US-PGPUB; DERWENT	2004/03/16 10:45
-	1107	(VoIP or "voice over IP") and ("load management" or (call near4 (admission or control)))	USPAT; US-PGPUB; DERWENT	2004/03/16 13:45
-	10	(VoIP or "voice over IP") and ("load management" or (admission near4 control)) and block$3 and capacity and (maximum with calls)	USPAT; US-PGPUB; DERWENT	2004/03/16 10:48
-	36	(VoIP or "voice over IP") and ("load management" or (admission near4 control)) and (cod$3 with frame$1)	USPAT; US-PGPUB; DERWENT	2004/03/16 10:50
-	32	((VoIP or "voice over IP") and ("load management" or (admission near4 control)) and (cod$3 with frame$1)) not (((VoIP or "voice over IP") and ("load management" or (admission near4 control)) and block$3 and capacity and (maximum with calls)) or ((VoIP or "voice over IP") and ("load management" or (admission near4 control)) and "call blocking"))	USPAT; US-PGPUB; DERWENT	2004/03/16 10:50
-	72	(VoIP or "voice over IP") and ("load management" or (call near4 (admission or control))) and (call near4 capacity)	USPAT; US-PGPUB; DERWENT	2004/03/16 10:52
-	66	((VoIP or "voice over IP") and ("load management" or (call near4 (admission or control))) and (call near4 capacity)) not (((VoIP or "voice over IP") and ("load management" or (admission near4 control)) and (cod$3 with frame$1)) or ((VoIP or "voice over IP") and ("load management" or (admission near4 control)) and block$3 and capacity and (maximum with calls)))	USPAT; US-PGPUB; DERWENT	2004/03/16 11:00
-	177	((VoIP or "voice over IP") and ("load management" or (admission near4 control))) not ((VoIP or "voice over IP") and ("load management" or (call near4 (admission or control))) and (call near4 capacity))	USPAT; US-PGPUB; DERWENT	2004/03/16 11:00
-	2	(VoIP or "voice over IP").ab. and ("load management" or (admission near4 control)) and (call near3 block$3) and capacity	USPAT; US-PGPUB; DERWENT	2004/03/16 11:05

-	14	(VoIP or "voice over IP").ab. and ("load management" or (admission near4 control))	USPAT; US-PGPUB; DERWENT	2004/03/16 11:29
-	46	(VoIP or "voice over IP" or gateway).ab. and "admission control"	USPAT; US-PGPUB; DERWENT	2004/03/16 11:29
-	42	((VoIP or "voice over IP" or gateway).ab. and "admission control") not (((VoIP or "voice over IP") and ("load management" or (admission near4 control))) and ((VoIP or "voice over IP") and ("load management" or (call near4 (admission or control))) and (call near4 capacity)))	USPAT; US-PGPUB; DERWENT	2004/03/16 13:44
-	58	(VoIP or "voice over IP") and ("load management" or (call near4 (admission or control)) with quality)	USPAT; US-PGPUB; DERWENT	2004/03/16 14:26
-	24	(VoIP or "voice over IP") and (call with (capacity or load or volume)) and (admission near5 control)	USPAT; US-PGPUB; DERWENT	2004/03/16 13:51
-	268	(VoIP or "voice over IP") and ((maximum or limit) with (call$1 or (number near2 call$1)))	USPAT; US-PGPUB; DERWENT	2004/03/16 14:27
-	152	(VoIP or "voice over IP") and ((maximum or limit) near4 (call$1 or (number near2 call$1)))	USPAT; US-PGPUB; DERWENT	2004/03/16 14:27
-	144	(VoIP or "voice over IP") and ((maximum or limit) near3 (call$1 or (number near2 call$1)))	USPAT; US-PGPUB; DERWENT	2004/03/17 08:37
-	11	(((VoIP or "voice over IP") and ((maximum or limit) near3 (call$1 or (number near2 call$1)))) and quality) and "admission control"	USPAT; US-PGPUB; DERWENT	2004/03/16 14:43
-	96	((VoIP or "voice over IP") and ((maximum or limit) near3 (call$1 or (number near2 call$1)))) and quality	USPAT; US-PGPUB; DERWENT	2004/03/16 14:43
-	58	(((VoIP or "voice over IP") and ((maximum or limit) near3 (call$1 or (number near2 call$1)))) and quality) and ((call or connect$3) near4 (accept$4 or reject$3))	USPAT; US-PGPUB; DERWENT	2004/03/16 14:44
-	9	(VoIP or "voice over IP") and (transcoder near1 "rate adaptation unit" or TRAU) with frame$1	USPAT; US-PGPUB; DERWENT	2004/03/17 11:16
-	2	(VoIP or "voice over IP") and ("frames per packet" with codec)	USPAT; US-PGPUB; DERWENT	2004/03/16 16:21
-	2	(VoIP or "voice over IP") and ("frames per packet" with cod$3)	USPAT; US-PGPUB; DERWENT	2004/03/16 16:22
-	18	(VoIP or "voice over IP") and ((frames near4 packet) with cod$3)	USPAT; US-PGPUB; DERWENT	2004/03/16 16:25
-	5	(VoIP or "voice over IP") and (capacity with (based near4 cod$3))	USPAT; US-PGPUB; DERWENT	2004/03/16 16:25
-	14	(VoIP or "voice over IP") and (((round adj trip adj delay) or latency) with (gateway$1 or gatekeeper)) and ping	USPAT; US-PGPUB; DERWENT	2004/03/17 08:52
-	0	(VoIP or "voice over IP") and (((round adj trip adj delay) or latency) with (admission adj control))	USPAT; US-PGPUB; DERWENT	2004/03/17 08:56
-	100	(VoIP or "voice over IP") and (((round adj trip adj delay) or latency) with (gateway$1 or gatekeeper))	USPAT; US-PGPUB; DERWENT	2004/03/17 08:53
-	400	(admission adj control) and ((round adj trip adj delay) or latency)	USPAT; US-PGPUB; DERWENT	2004/03/17 08:56
-	12	(admission adj control) with ((round adj trip adj delay) or latency)	USPAT; US-PGPUB; DERWENT	2004/03/17 08:56

US 20040008627A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2004/0008627 A1

Garg et al. (43) Pub. Date: Jan. 15, 2004

(54) METHOD AND APPARATUS FOR PERFORMING ADMISSION CONTROL IN A COMMUNICATION NETWORK

(76) Inventors: **Sachin Garg**, Green Brook, NJ (US); **Martin Kappes**, Bridgewater, NJ (US)

Correspondence Address:
Ryan, Mason & Lewis, LLP
Suite 205
1300 Post Road
Fairfield, CT 06430 (US)

(21) Appl. No.: **10/261,243**

(22) Filed: **Sep. 30, 2002**

Related U.S. Application Data

(60) Provisional application No. 60/395,445, filed on Jul. 12, 2002.

Publication Classification

(51) **Int. Cl.**[7] **H04J 1/16**; H04J 3/16

(52) **U.S. Cl.** **370/235**; 370/468

(57) **ABSTRACT**

A method and apparatus are disclosed for assessing the available resources in a network and using the assessment for admission control. A VoIP call can be established with a device only if the network has sufficient resources to accommodate the call or it is possible to make such resources available by curtailing ongoing data connections. A network utilization characteristic (NUC) provides a measure of network capacity. The network utilization characteristic of a flow is the fraction of time the network is busy transmitting data for that flow. The sum of the network utilization characteristics of all flows yields the fraction of time the network is busy transmitting data. The difference between one and the sum of all flows indicates the time that the network is idle in the measured time interval. A new flow can be accommodated if the NUC of the new flow is smaller than this difference value.

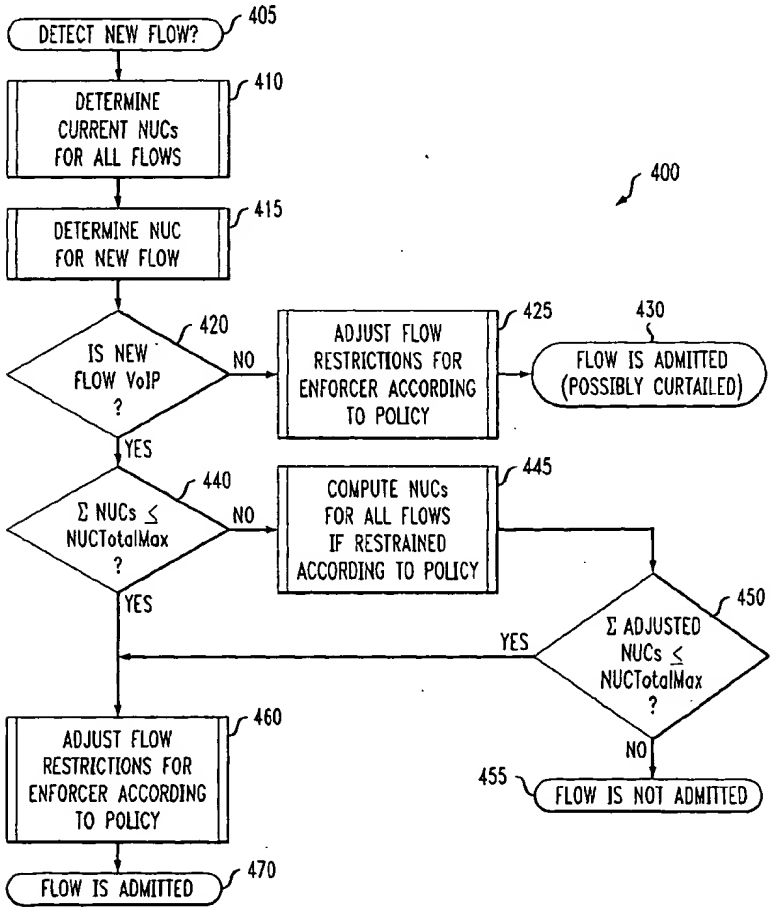
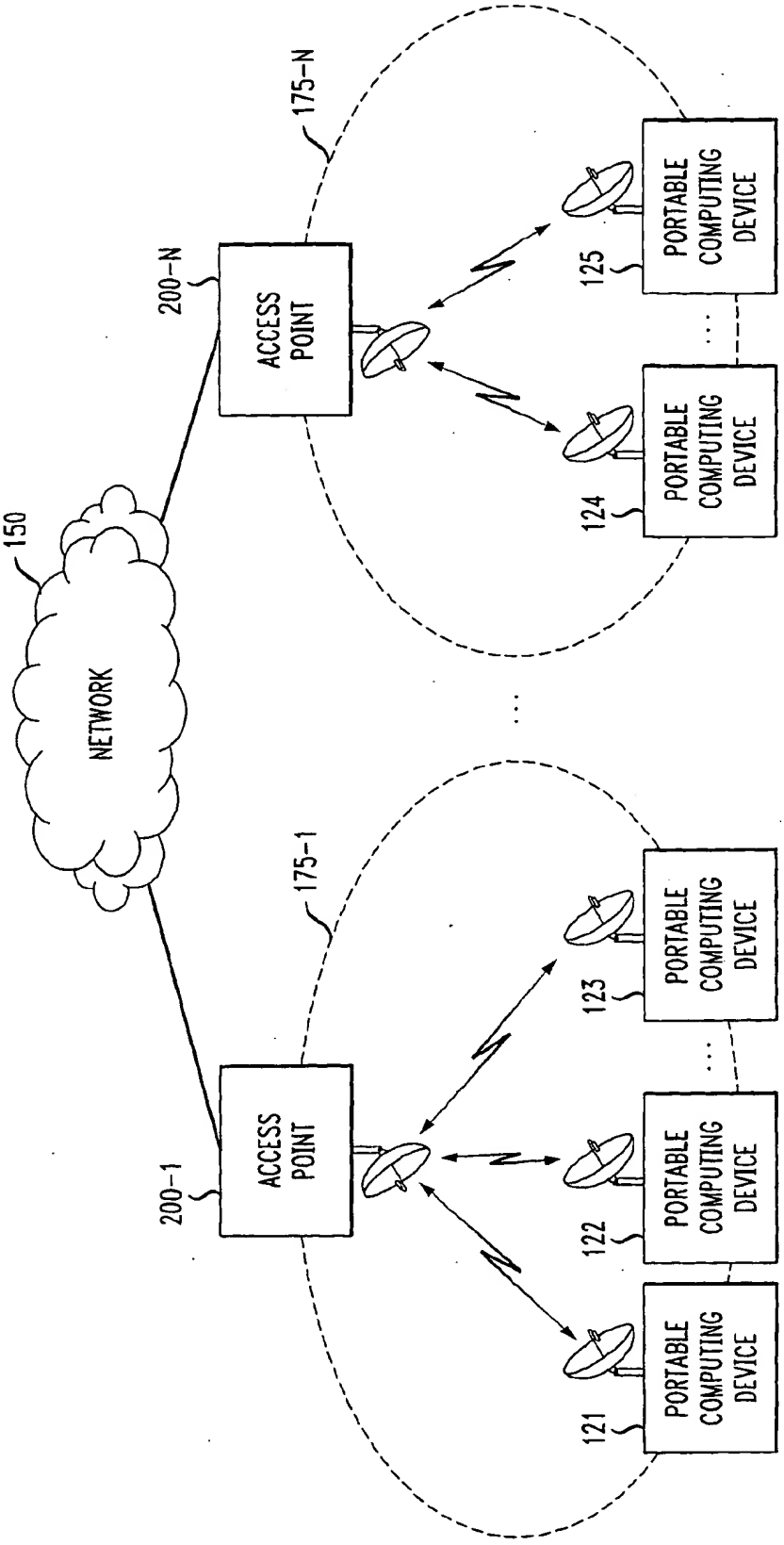


FIG. 1


NETWORK
150
ACCESS POINT
200-N
ACCESS POINT
200-1
175-N
175-1
PORTABLE COMPUTING DEVICE
125
PORTABLE COMPUTING DEVICE
124
PORTABLE COMPUTING DEVICE
123
PORTABLE COMPUTING DEVICE
122
PORTABLE COMPUTING DEVICE
121

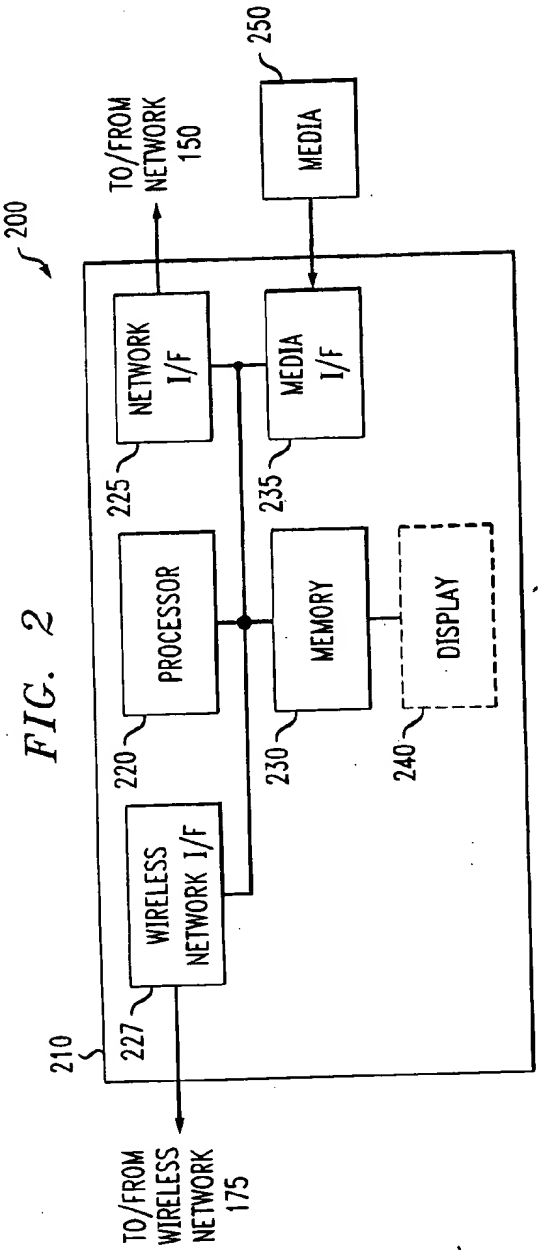

200
TO/FROM NETWORK 150
250
MEDIA
NETWORK I/F
MEDIA I/F
225
235
PROCESSOR
MEMORY
DISPLAY
220
230
240
WIRELESS NETWORK I/F
227
210
TO/FROM WIRELESS NETWORK 175

FIG. 2

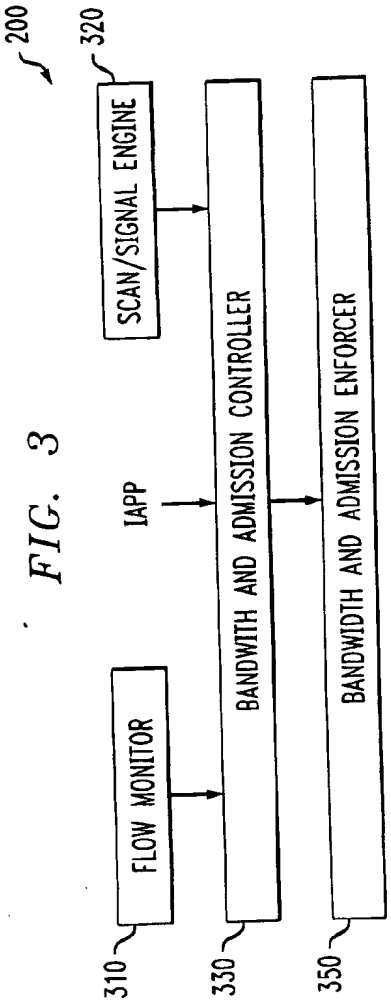
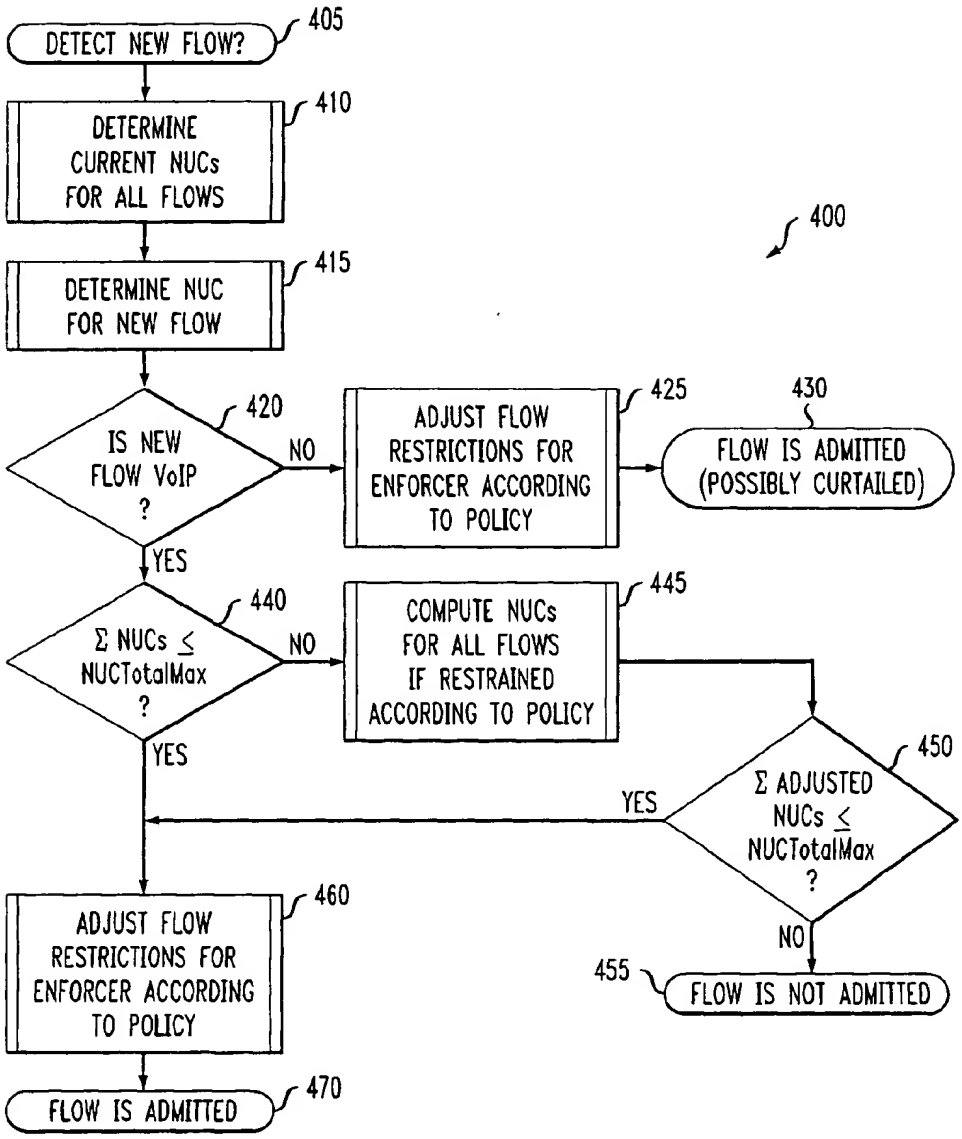

200
320
SCAN/SIGNAL ENGINE
IAPP
FLOW MONITOR
BANDWITH AND ADMISSION CONTROLLER
BANDWIDTH AND ADMISSION ENFORCER
310
330
350

FIG. 3

FIG. 4


DETECT NEW FLOW?
405
DETERMINE CURRENT NUCs FOR ALL FLOWS
410
400
DETERMINE NUC FOR NEW FLOW
415
IS NEW FLOW VoIP ?
420
NO
ADJUST FLOW RESTRICTIONS FOR ENFORCER ACCORDING TO POLICY
425
430
FLOW IS ADMITTED (POSSIBLY CURTAILED)
YES
Σ NUCs ≤ NUCTotalMax ?
440
NO
COMPUTE NUCs FOR ALL FLOWS IF RESTRAINED ACCORDING TO POLICY
445
YES
Σ ADJUSTED NUCs ≤ NUCTotalMax ?
450
YES
NO
455
FLOW IS NOT ADMITTED
ADJUST FLOW RESTRICTIONS FOR ENFORCER ACCORDING TO POLICY
460
FLOW IS ADMITTED
470

METHOD AND APPARATUS FOR PERFORMING ADMISSION CONTROL IN A COMMUNICATION NETWORK

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 60/395,445, filed Jul. 12, 2002.

FIELD OF THE INVENTION

[0002] The present invention relates to admission control techniques, and more particularly, to admission control techniques for voice and data traffic in communication networks, such as communication networks based on the IEEE 802.11b standard.

BACKGROUND OF THE INVENTION

[0003] Wireless networks based on the IEEE 802.11b standard are increasingly deployed in corporate enterprises and public hot spots, such as airports, hotels and conference facilities. Generally, wireless networks based on the IEEE 802.11b standard are used primarily to provide wireless data access from portable computing devices, such as portable computers and personal digital assistants (PDAs), to a wired network, such as an enterprise network or the Internet. Increasingly, wireless networks based on the IEEE 802.11b standard are being used to carry voice traffic, typically referred to as "voice over IP" (VoIP) traffic.

[0004] The IEEE 802.11b standard has a large fixed overhead. Thus, the bandwidth available at the payload sizes typical for VoIP traffic is far less than the bandwidth available when using the network for data traffic. The 802.11b standard currently supports data rates of 1, 2, 5.5 and 11 Mbps. When sending data frames with the maximal data rate of 11 Mbps in such an IEEE 802.11b network, the maximal achievable throughput is approximately 6.2 Mbps. When sending frames for VoIP traffic, however, the maximal achievable throughput is only approximately 2 Mbps, for a typical audio payload size for a packet in a Real Time Protocol (RTP) audio stream.

[0005] The significant difference between the achievable throughputs for audio traffic and VoIP traffic is primarily due to the large transmission overhead per frame, regardless of the frame size. Depending on the actual average transmission rate, the number of simultaneous VoIP calls in a cell (the term "cell" is used to refer to the Basic Service Set (BSS) using terminology of the 802.11 standard) of the wireless network is between 4 and 17 calls for an exemplary G711 codec with 30 milliseconds of audio data per packet.

[0006] If a wireless network based on the IEEE 802.11b standard accepts one additional call or one additional data connection that exceeds the capacity of the wireless network, an unacceptable call quality for all ongoing VoIP calls results. Furthermore, if the load offered to the network is higher than the capacity of the network, the Distributed Coordination Function (DCF) medium access scheme of the IEEE 802.11b standard curtails the client with the highest load first. In most cases, the access point of the wireless cell provides more traffic to the network than the associated stations. Hence, the access point gets curtailed first which leads to unacceptable packet loss for all VoIP streams transmitted from the access point to a client resulting in poor call quality for all connections. Thus, a need exists for a technique for measuring network capacity and for providing admission control in a wireless network carrying VoIP traffic.

SUMMARY OF THE INVENTION

[0007] Generally, a method and apparatus are disclosed for assessing the available resources in a network and to thereafter use the assessment for admission control. The disclosed admission control techniques allows a VoIP call to be established to or from a device only if the network has sufficient resources to accommodate the call or it is possible to make such resources available by curtailing ongoing data connections. A network utilization characteristic (NUC) is measured on a per-flow basis to measure network capacity.

[0008] The network utilization characteristic of a flow is defined as the fraction of time the network is busy transmitting data for that flow. Two flows having the same network utilization characteristic are said to be bandwidth equivalent. The sum of the network utilization characteristics of all flows (including auxiliary flows) in the network yields the fraction of time the network is busy transmitting the data for all flows and is referred to herein as NUCTotal. If the medium is constantly busy transmitting data of the flows, then NUCTotal equals one. Consequently, the difference between one and the sum of all flows (NUCTotal) indicates the time that the network is idle in the measured time interval. Generally, a new flow can be accommodated without sacrificing the quality of other flows if the NUC of the new flow is smaller than this difference value.

[0009] A disclosed admission control process (i) classifies traffic into flows; (ii) categorizes and measures the network utilization characteristics of each flow;(iii) controls the network usage of each flow to the desired value in accordance with the quality of service (QoS) requirements for all flows; and (iv) enforces admission control based on the predicted network utilization characteristic of the new flow and the current network utilization characteristic of all flows. Once a new flow is detected, the admission control process computes the current NUCs for all existing flows (NUCTotal), and estimates the NUC for the new flow. Generally, if NUCTotal plus the NUC for the new flow is less than a predefined upper limit, NUCTotalMax, then the new flow will be admitted. If NUCTotal plus the NUC for the new flow is not less than the predefined upper limit, it is determined whether other flows are to be curtailed for the new flow, based on the established policies for the network, or whether the new flow should be rejected. If NUCTotal plus the NUC for the new flow is less than the predefined upper limit, then the flow restrictions are calculated and enforced and the new flow is admitted.

[0010] Theoretically, the predefined upper limit, NUCTotalMax, can be set to one. According to another aspect of the invention, the predefined upper limit, NUCTotalMax, may be used to (i) account for certain inaccuracies that might arise if data needed to determine the NUC of flows and auxiliary flows is only partially collected, or (ii) create a network capacity backup that might be needed in some situations (or both).

[0011] A more complete understanding of the present invention, as well as further features and advantages of the

present invention, will be obtained by reference to the following detailed description and drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] **FIG. 1** illustrates a network environment in which the present invention can operate;

[0013] **FIG. 2** is a block diagram illustrating an exemplary embodiment of an access point of **FIG. 1**;

[0014] **FIG. 3** is a schematic block diagram illustrating the various components of the access point of **FIG. 2** that are used for admission control in accordance with the present invention; and

[0015] **FIG. 4** is a flow chart describing an exemplary implementation of an admission control process performed by the access point of **FIG. 2**.

DETAILED DESCRIPTION

[0016] **FIG. 1** illustrates a network environment **100** in which the present invention can operate. As shown in **FIG. 1**, one or more access points **200-1** through **200-N**, discussed below in conjunction with **FIG. 2**, provides wireless data access from one or more portable computing devices **121-125**, such as portable computers and personal digital assistants (PDAs), to a network **150**, such as an enterprise network, a public telephone network or the Internet (or a combination of the foregoing). The portable computing devices **121-125** generate data or voice traffic (or both). The network **150** can include wired and wireless components. Each access point **200**-n defines a corresponding wireless network **175**-n, as shown in **FIG. 1**. The access point **200** is thus often referred to as a wireless gateway.

[0017] According to one aspect of the invention, the access point **200** employs a network utilization characteristic (NUC) to assess the available resources in the wireless network **175**-n and to thereafter use the NUC assessment for admission control. The present invention allows a VoIP call to be established to or from a portable computing device **121-125** only if the network **175**-n has sufficient resources to accommodate the call or it is possible to make such resources available by curtailing ongoing data connections.

[0018] **FIG. 2** is a block diagram illustrating an exemplary access point **200**. As shown in **FIG. 2**, the access point **200** comprises a computer system **210** that optionally interacts with a Digital Versatile Disk (DVD) **250**. Computer system **210** comprises a processor **220**, a network interface **225**, a wireless network interface **227**, a memory **230**, a media interface **235**, and an optional display **240**. Network interface **225** allows computer system **210** to connect to network **150**, wireless network interface **227** allows computer system **210** to connect to a wireless network **175**, while media interfaces **235** allows computer system **235** to interact with media such as a hard drive or DVD **250**.

[0019] As is known in the art, the methods and apparatus discussed herein may be distributed as an article of manufacture that itself comprises a computer-readable medium having computer-readable code means embodied thereon. The computer-readable program code means is operable, in conjunction with a computer system such as computer system **210**, to carry out all or some of the steps to perform the methods or create the apparatuses discussed herein. The computer-readable medium may be a recordable medium (e.g., floppy disks, hard drives, optical disks such as DVD **250**, or memory cards) or may be a transmission medium (e.g., a network comprising fiber-optics, the world-wide web, cables, or a wireless channel using time-division multiple access, code-division multiple access, or other radio-frequency channel). Any medium known or developed that can store information suitable for use with a computer system may be used. The computer-readable code means is any mechanism for allowing a computer to read instructions and data, such as magnetic variations on a magnetic medium or height variations on the surface of a compact disk, such as DVD **250**.

[0020] Memory **230** configures the processor **220** to implement the methods, steps, and functions disclosed herein. The memory **230** could be distributed or local and the processor **220** could be distributed or singular. The memory **230** could be implemented as an electrical, magnetic or optical memory, or any combination of these or other types of storage devices. Moreover, the term "memory" should be construed broadly enough to encompass any information able to be read from or written to an address in the addressable space accessed by processor **210**. With this definition, information on a network, accessible through network interface **225** or **227**, is still within memory **230** because the processor **220** can retrieve the information from the network. It should be noted that each distributed processor that makes up processor **220** generally contains its own addressable memory space. It should also be noted that some or all of computer system **210** could be incorporated into an application-specific or general-use integrated circuit.

[0021] Optional video display **240** is any type of video display suitable for interacting with a human user of system **200**. Generally, video display **240** is a computer monitor or other similar video display.

[0022] As discussed further below in conjunction with **FIG. 3**, the access point **200** performs an admission control process **400** that includes the following tasks: (i) classifying traffic into flows at some granularity; (ii) categorizing and measuring the network utilization characteristics of each flow; (iii) controlling the network usage of each flow to the desired value in accordance with the quality of service (QoS) requirements for all flows; and (iv) enforcing admission control based on the predicted network utilization characteristic of the new flow and the current network utilization characteristic of all flows.

[0023] It is noted that while the admission control process **400** is illustrated as being performed by the access point **200**, the admission control process **400** may alternatively be performed by a switch that connect the access point **200** to the network **150** or some device between the access point **200** and the switch. In addition, the functionality may be split among a number of such devices. In addition, while the present invention is illustrated in the context of a wireless network, the present invention applies to both wired and wireless networks. Finally, while the present invention is focused on the use of NUCs for admission control for VoIP in wireless networks, the actual policies for enforcing such admission control are not within the scope of the invention.

Flows

[0024] The access point **200** partitions the traffic into flows. Each transmitted frame belongs to exactly one flow.

The criterion could be frame/packet identifiers at various network layers starting from the MAC Layer (Layer 2) up to the Transport Layer (Layer 4). A frame is classified based on a lower layer criteria only if the higher layer information is unavailable indicating that the packet/frame belongs to a specific layer. For example, Internet Control Message Protocol (ICMP) packets are Layer-3 packets. Furthermore, certain frames such as Beacon, probe-request and response frames are limited only to Layer-2. Generally, higher layer criteria, if possible, provides finer granularity of classification.

[0025] A Transport Layer flow is uniquely determined by the four-tuple (sourceIP, dstIP, srcPort, dstPort) and by the Transport Layer protocol type of the flow, namely, Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP). Given two communicating stations, this classification scheme implies that the network utilization is measured separately for each traffic direction. A TCP connection results in two TCP flows and a VoIP call consists of two UDP flows. In other words, this separation captures the asymmetric nature of most data connections, such as file/web downloads in terms of network utilization. Apart from flows associated with voice and data traffic, auxiliary flows are also considered. Auxiliary flows represent network activities such as erroneous transmissions and collisions that cannot be accredited to any particular flow but typically represent wasted network capacity.

[0026] VoIP Flows and Data Flows

[0027] Although the admission control scheme of the present invention could be extended to address other flow categories, the exemplary embodiment described herein considers two classes of flows, namely VoIP flows and data flows.

[0028] VoIP traffic flows consist of real-time audio data. Such flows typically use connection-less transport layer protocols, such as the UDP protocol, that do not guarantee the delivery of transmitted audio data information (i.e., lost packets are not retransmitted). Furthermore, the amount of traffic that the sender is transmitting to the receiver is typically fixed for the duration of the flow. Thus, it will not be influenced by parameters such as current load conditions of the network, as indicated by, e.g., lost packets or ICMP messages. Although occasionally losing some packets does not render the flow useless, the loss of more than 1% of the data usually deteriorates the quality of the flow to an unacceptable level. Thus, curtailing such a flow by dropping a certain percentage of the packets renders the whole flow useless. Therefore, such a flow must either be given the full bandwidth it requires or be completely dropped. The bandwidth such a flow needs can be easily detected either by transparently examining the messages exchanged between the endpoints of the flow during connection setup time or by just observing the traffic patterns.

[0029] As the average transmission speed of all clients in the wireless network and the average actual back-off are known, the NUC for that flow can be accurately estimated. It should be noted that while the number of frames and the size of the transmitted frames remains constant during the lifetime of the flow, the NUC of such a flow can change significantly due to variations in the transmission data rate, especially in the case of mobile clients, as discussed further below.

[0030] Data flows, on the other hand, use connection-oriented transport layer protocols, such as TCP. In most cases, such protocols also provide flow control and congestion control. Apart from resending lost data, flow control and congestion control adapt the network usage of the connection due to current conditions, such as available buffer size on the receiving side or network congestion. In other words, the bandwidth used by such a flow, as well as the NUC of the flow, can change over time. While the user may notice a smaller bandwidth, e.g., by a longer transmission time for a file or web page, a reduction in bandwidth does not render the flow useless. Therefore, such flows can be throttled down, for example, by means of TCP congestion control. The use of TCP congestion control and other mechanisms to curtail the bandwidth of such a flow is described in S. Garg et al., "Wireless Access Server for Quality of Service and Location Based Access Control in 802.11 Networks," Proc. of the Seventh IEEE Symposium on Computers and Communication (ISCC'02), Taormina, Italy, 2002, incorporated by reference above. As opposed to VoIP flows, the bandwidth of a data flow cannot be assessed from the values collected when the flow is detected. In the following discussion, it is assumed that no information about the NUC of the flow is available.

Network Utilization Characteristic

[0031] As previously indicated, the disclosed "network utilization characteristic" provides an accurate assessment of the available resources in the wireless network. It has been observed that bandwidth is an inappropriate criterion for measuring network utilization of wireless networks. In an 802.11b wireless infrastructure based network with a single client, the fixed overhead per frame transmission is 765 μs when transmitting at 11 Mbps. The time it takes to transmit 100 bytes at this data rate is 72.7 μs. Consequently, the transmission of 1000 bytes takes 727 μs. Therefore, when transmitting 100 byte frames, the transmission of a single 100 byte frame including all overhead takes 837.5 μs and thus the maximum number of frames that can be transmitted per second is 1193, resulting in a maximal throughput of 954 Kbps. For 1000 bytes size per frame, a maximum of 670 frames can be transmitted per second, resulting in a maximal throughput of 5.36 Mbps.

[0032] Thus, bandwidth provides an inappropriate criterion for determining whether a new VoIP stream or data stream can be accommodated without exceeding the capacity of a wireless network. The situation becomes even more complex when multiple flows are to be considered as necessary for admission control.

[0033] According to one aspect of the present invention, a network utilization characteristic is measured on a per-flow basis to measure network capacity. The network utilization characteristic (NUC) of a flow is defined as the fraction of time per time unit needed to transmit the flow over the network. As used herein, the term "network utilization characteristic" of a flow shall also include any method of measuring the channel usage of a communication network that produces the same value for two flows that use the same fraction of time per time unit to transmit on the channel and produces the same ranking of flows with respect to their network usage as the NUC measure. For example, such measures include the reciprocal value of the NUC as defined above, the logarithm of the NUC as defined above, or the

size of a flow in Mbps that uses frames with 1472 bytes of payload only that could be transmitted in the same fraction of time per time unit as the flow under consideration.

[0034] Two flows having the same network utilization characteristic are said to be bandwidth equivalent. In the exemplary embodiment described herein, considerations are based on a "per second" basis. Any other interval, such as a beacon period, can be used without any change in the results, as would be apparent to a person of ordinary skill in the art.

[0035] Consider a flow with 100 byte size frames having a bandwidth of 100 Kbps and transmission parameters as outlined above. The time (overhead and actual data transmission time) to transmit a single frame is 837 μs. The time to transmit 100 Kbps using a frame size of 100 bytes requires 125 packets per second. The time to transmit 125 packets of that size takes 104.6 ms. Thus, the NUC of the flow is 0.1046. Now consider a flow with 1000 byte size frames having a bandwidth of 1 Mbps and the transmission parameters as outlined above. The flow sends 125 frames per second and the transmission time for a single frame is 1492 μs. Hence, the NUC of the flow is 0.1865.

[0036] The sum of the NUCs of all flows (including auxiliary flows) in the network yields the fraction of time the network is busy. Consequently, the difference between one and the sum of all flows indicates the time that the medium is idle. Generally, a flow can be accommodated without sacrificing other flows if its NUC is going to be smaller than this difference value.

Measuring Network Utilization Characteristic

[0037] As discussed hereinafter, the data needed to compute (or accurately assess) the network utilization characteristic (NUC) of a flow is readily available in the access-point. Specifically, two parameters are needed for determining the NUC of a flow, namely, the number of frames sent for that flow per second and the average transmission time for a frame.

[0038] For a detailed discussion of the monitoring of packets per second, used bandwidth and average packet size per flow, see S. Garg et al., "Wireless Access Server for Quality of Service and Location Based Access Control in 802.11 Networks," Proc. of the Seventh IEEE Symposium on Computers and Communication (ISCC'02), Taormina, Italy, 2002, incorporated by reference herein. For regular Ethernet, these three parameters would be sufficient to compute the transmission time per packet. For 802.11 networks, however, the overhead due to the channel access mechanism is not captured in any of the parameters. In fact, this overhead is substantial and cannot be ignored.

[0039] As indicated above, the IEEE 802.11, 802.11a and 802.11b standards for wireless Local Area Networks, described at http://standards.ieee.org/getieee802/802.11.html, include a Collision Avoidance (CSMA/CA) medium access scheme according to the Distributed Coordination Function (DCF) that helps to explain the associated overhead. The MAC protocol is designed to prevent collisions from occurring. Furthermore, unicast frames are acknowledged by the receiving station. The acknowledgment (ACK) is sent out after the transmission has finished and a defined short inter frame spacing (SIFS) time period has elapsed. If a node wants to transmit a frame and senses that the medium is idle for a distributed coordination function inter frame spacing (DIFS) time period, the node may start transmitting. Since DIFS is longer than SIFS, a correctly received frame can always be acknowledged before the next frame is transmitted.

[0040] If a node wants to start transmitting while the medium is busy or if it wants to transmit another frame after just finishing a transmission, the node also waits for the medium to be idle for the DIFS period. Then, the node does not begin to transmit immediately but enters a contention phase for the medium. Contention is performed by choosing an integer random back-off interval from a certain interval. The random back-off interval determines the number of time slots the client defers its transmission in addition to the DIFS time.

[0041] If the medium is determined to be idle in such a "slot," the back-off timer is decreased by one. If the random back-off timer has decreased to zero, the node starts transmitting. If another node starts transmitting before this happens, the node continues to count down an amount equal to the back-off timer after the medium has been sensed idle for the DIFS period. Thus, if multiple clients want to transmit a frame, the one with the lowest random back-off time will win the contention for the medium.

Determining Transmission Time of a Frame

[0042] Transmission time of a frame consists of time to transmit the frame itself, PHY layer header overhead, overhead of an ACK and overhead imposed by the DCF mechanism. The DCF overhead consists of one SIFS, one DIFS and a back-off interval. The transmission of ACKs and data frames causes a physical layer overhead of 192 μs in each case. The duration of sending the data depends on the frame length and the transmission speed. For an ACK, the size is fixed at 14 bytes so the duration only depends on the transmission speed. In addition, the SIFS and DIFS intervals are fixed. R denotes the transmission speed in bps and b denotes the size of the data frame in bytes. Apart from frame size, the only information needed to calculate the tabulated values is the transmission speed. The time needed for transmitting the frame includes the following components:

Data Frame-	192 μs + b · 8/R
SIFS-	10 μs
ACK-	192 μs + 14 · 8/R
DIFS-	50 μs

[0043] The component of the transmission time not yet addressed is the back-off value. In wireless mediums, the actual number of idle back-off slots immediately preceding the transmission is of more interest than the actual back-off window that was chosen for the transmitted frame. A slot time in 802.11b is 20 μs, so the number of slots waited between the end of DIFS and the transmission multiplied by the slot time yields the desired value.

Determining the NUC of a Flow

[0044] To accurately determine the NUC of a flow, it is sufficient to compute the transmission time of all frames transmitted that belong to the flow and then sum up these

values on a per-second basis. However, a simplified methodology for computing the NUC of a flow is initially presented. If the number of frames sent, n; the average number of bytes sent per data frame, b; the average transmission speed (on a per-byte basis) R_{avg}; and the average number of actual back-off intervals waited before transmission, s, are given, the NUC can be computed as follows:

[0045] $NUC=n\cdot(s\cdot 20\ \mu s+192\ \mu s+(b\cdot 8)/R_{avg}+10\ \mu s+192\ \mu s+(14\cdot 8)/R_{avg}+50\ \mu s)$.

[0046] It is noted that the NUC obtained is identical with the number that would be computed by summing up the transmission time needed for all frames of the flow.

[0047] While the number of frames sent per second and the average number of bytes sent per data frame can be obtained by standard means (such as SNMP MIBs), the transmission speed can be observed by both the receiving and the sending party. As the admission control scheme is most likely to be implemented in the access point and as all traffic in an infrastructure-based network either origins from the access point or is destined to the access point, it is feasible to get accurate information about the actual transmission speed of a particular frame and thus also of the average transmission speed of a particular flow. The transmission speed is determined by the wireless station based on factors such as frame error rates and strength of the radio signal, and is not dependent on the characteristics of a flow.

[0048] In fact, all flows emanating from a station will use the same transmission speed at a given time. Therefore, it suffices to measure this parameter on a per-station basis and use the value for all flows from that station. Moreover, instead of recording the transmission speed of every frame, it may be sufficient to compute the average transmission speed based on sampling frames in a short period such as a beacon-period.

[0049] Determining the actual back-off interval before transmission on a per-frame basis is not possible for anyone but the station transmitting the frame. An observer on the channel cannot distinguish between idle times caused on the channel because of back-off slots before a transmission versus the idle time caused because the station did not attempt to transmit at all. However, an average back-off value on a per-second basis is sufficient for computing the network utilization characteristic. Due to the fair nature of DCF, the average back-off interval experienced by any station is the same. In other words, the average back-off interval can be measured at the access point and the same value can be used for all stations. As previously indicated, it may suffice to measure the average back-off interval based on a few random samples at the access-point in a fixed duration.

[0050] As the preceding discussion shows, the values necessary to determine the NUC of a flow can be easily derived by standard means or accurately estimated even without having full access to the PHY/MAC layers of the access point **200**. In fact, it is believed that there is a spectrum of possibilities for trading off between accuracy and simplicity of data collection. For example, in order to assess the aggregated NUC of all flows in the network, it is sufficient to know the number of packets sent, the bytes sent, the average transmission speed in the network and the average actual back-off. While only the computation or estimation of the NUC of flows has been addressed, it is also apparent that the data needed to compute the NUC of auxiliary flows, for instance due to collisions or erroneous transmissions, is present in the access point. For other ways of accounting for such traffic, see the section below entitled "Uses for NUCTotalMax."

Using NUCs for Admission Control

[0051] In the following discussion, the use of network utilization characteristics for admission control for VoIP traffic in wireless networks is described. As previously indicated, the network utilization characteristic of a flow is the fraction of time the network is busy transmitting data for that flow. The sum of all network utilization characteristics of all flows (including auxiliary flows) is the fraction of time the network is busy transmitting the data for all flows and is referred to herein as NUCTotal. If the medium is constantly busy transmitting data of the flows, then NUCTotal equals one. Hence, the difference between one and NUCTotal is the fraction of time the medium has been idle in the measured time interval.

[0052] The NUC of a flow can get very close to one. A network utilization characteristic of 0.993 was achieved for a single flow of UDP data constantly sending out frames to the network in a single client scenario. The difference to one is due to MAC-Level exchange of information such as beacons and other management frames which is captured in other flows.

[0053] Although it is possible to measure the NUC of Layer 2 traffic and of collisions and erroneous transmissions, these network activities can also be taken care of by defining an adjustable parameter NUCTotalMax that can range from zero to one and would typically be slightly less than one, as discussed below in the section entitled "Uses for NUCTotalMax." This parameter defines the de-facto limit of the network. By adjusting this parameter, the present invention can adapt either for unusual situations such as extremely high frame loss due to interference or it could be used to adjust the scheme in case the NUCs for flows are not accurately computed but estimated. In fact, whereas an accurate computation of the NUC of a flow is feasible, it is anticipated that most systems would trade off accuracy against other factors.

[0054] As used herein, a new flow is a flow that is new to the wireless network. A new flow need not be one that has just been established but could be an ongoing flow from or to a client that has just roamed into this cell of the wireless network. It is noted that a flow can be detected by observing the first packet from that flow. This holds for a flow that is about to be established as well as an ongoing flow that just roamed into the cell. VoIP streams can be detected before they are actually established by monitoring for traffic to initiate the call, such as packets directed to a H.323 port or packets containing SIP protocol messages. Similarly, the start of TCP flows can be detected by examining the SYN/ACK bits in the TCP packet header. Generally, a flow cannot create a significant load on the network before it is detected.

[0055] Clearly, calculating the NUCs as described above generates accurate data regarding the usage of the network in a past interval. The present invention assumes that the past NUC of a flow constitutes a good estimate of the flows future NUC. In other words, a steady state usage model is

assumed for computing the permissibility of a new VoIP flow. Since the bandwidth used by a flow may change over time, it is necessary to also enforce bandwidth restrictions of non-VoIP flows in order to provide VoIP admission control.

[0056] This holds especially true if the network operates close to its capacity limits. Along the same lines, when a new data flow is detected, the NUCavailable for this flow is to be determined and enforced depending on the policies in the system. While the NUC of a flow accurately measures the network resources used by the flow, bandwidth is a variable that will probably also be considered when formulating policies for admission control. Apart from determining whether new flows should be admitted, the NUCs of all flows need to be monitored constantly.

Access Point **200**

[0057] **FIG. 3** is a schematic block diagram illustrating the various components of the access point **200** that are used for admission control. Generally, the control of network utilization is a generalized form of bandwidth control. For a more detailed discussion of techniques to monitor and control the bandwidth consumed by each such flow, see, for example, U.S. patent application Ser. No. 10/178,762, filed Jun. 25, 2002, entitled "System and Method for Providing Bandwidth Management for VPNs," incorporated by reference herein. Network usage can be enforced using the same techniques as bandwidth usage. The target numbers and the measurement of actual usage are based, however, on the network utilization characteristic in accordance with the present invention.

[0058] As shown in **FIG. 3**, the access point **200** includes a flow monitor **310** that monitors the traffic to and from the wireless network, classifies the traffic into flows and computes the bandwidth of each flow. A scan/signal engine **320** detects traffic intended to establish a VoIP flow using various means, for instance by leveraging Resource Reservation Protocol (RSVP), discussed below, if the network is admission control enabled or by analyzing the traffic in standalone mode, as discussed further below. A new flow or intention to establish a new flow can be detected by (i) the scan/signal engine **320** before the flow is about to be established; (ii) the flow monitor **310** when the flow has been started; or (iii) by leveraging the Inter Access Point Protocol (IAPP) in case of a client roaming into a cell controlled by the access point **200** with an ongoing VoIP call.

[0059] When a new flow or an intention to establish a new flow is detected, a bandwidth admission controller **330** is engaged and the steps followed to determine whether admission control is needed are identical. However, when it is determined that admission control is needed, the manner in which such admission control is enforced by a bandwidth and admission enforcer **350** is different and varies in accordance with established network policies. The operation of the bandwidth admission controller **330** is discussed hereinafter in conjunction with **FIG. 4**.

[0060] **FIG. 4** is a flow chart describing an exemplary implementation of an admission control process **400**. As previously indicated, the admission control process **400** the access point **200** performs the admission control process **400** that includes the following tasks: (i) classifying traffic into flows at some granularity; (ii) categorizing and measuring the network utilization characteristics of each flow; (iii) controlling the network usage of each flow to the desired value in accordance with the quality of service (QoS) requirements for all flows; and (iv) enforcing admission control based on the predicted network utilization characteristic of the new flow and the current network utilization characteristic of all flows.

[0061] As shown in **FIG. 4**, the admission control process **400** is initiated during step **405** when a new flow is detected. As indicated above, the access point **200** classifies incoming and outgoing traffic into flows. If possible, the access point **200** looks inside the IP packets and determines if the higher-layer protocol is TCP or UDP. In this case, the flow is typically defined by the four tuple <srcIP, dstIP, srcPort, dstPort>. Each incoming packet is examined for these four values, and based on the results, is classified as belonging to an existing flow or a new flow. TCP rate control mechanisms have been used to monitor and control the bandwidth consumed by each such flow. For a more detailed discussion of techniques to monitor and control the bandwidth consumed by each such flow, see, for example, U.S. patent application Ser. No. _____, filed _____, entitled "Bandwidth Management for VPNs over Wireless Local Area Networks using TCP Congestion Control," incorporated by reference herein.

[0062] Once a new flow is detected, the admission control process **400** computes the current NUCs for all existing flows during step **410** (NUCTotal), and estimates the NUC for the new flow during step **415**. The access point **200** monitors the network utilization characteristics of each flow (given a particular method of flow classification), for example, by taking a fixed or sliding time window and then counting the number of packets and their size belonging to a particular flow that arrive in that time window. For each flow, the following characteristics are collected in a fixed or sliding time window: number of packets; average size of packets in bytes or bits; and average transmission speed (averaged on a per bit or byte basis). These values are used to compute the network utilization characteristics (NUC) of each flow as outlined above. Intuitively, the NUC for each flow captures the fraction of time per time unit that is necessary to transmit each flow over the wireless link. The NUC of each flow can be computed with the above values and the average random back-off value for medium access.

[0063] Generally, if NUCTotal plus the NUC for the new flow is less than NUCTotalMax (step **440**), a VoIP flow will be admitted. If not, a decision whether other flows are to be curtailed for the new flow is made, based on the established policies. If so, the restrictions are calculated and enforced and the new flow is admitted. If not, the new flow is not admitted.

[0064] In the exemplary embodiment, a test is performed during step **420** to determine if the new flow is a VoIP flow or a data flow. If no other information is available, the following characteristics can be used to distinguish VoIP from non-VoIP streams with a high degree of reliability:

[0065] the transport protocol is UDP;

[0066] the bandwidth used does not vary over time;

[0067] the bandwidth used is between 20-80 Kbps; or

[0068] if there is a flow from a machine A to B, there is also a flow in the opposite direction from machine

B to A. Furthermore, if the transport protocol is UDP and the real time protocol (RTP) is employed, a new flow can be classified as a VoIP flow and the specific codec that is employed may be determined.

[0069] Although this method will in most cases work reliably for flow classification, it is better to proactively detect a VoIP flow by either monitoring signaling data prior to flow establishment such as H.323 or SIP messages as discussed above.

[0070] If it is determined during step **420** that the new flow is a data flow, then program control proceeds to step **425** where the flow restrictions are adjusted according to the network policy. The flow is admitted during step **430** although it may not be allocated the full desired bandwidth. As previously indicated, a reduction in bandwidth does not render a data flow useless, although the user may notice a smaller bandwidth, e.g., by a longer transmission time for a file or web page.

[0071] If, however, it is determined during step **420** that the new flow is a VoIP flow, then program control proceeds to step **440** where a further test is performed during step **440** to determine if the sum of all NUCs for the existing and new flow(s) (NUCTotal) is less than NUCTotalMax. If it is determined during step **440** that NUCTotal is not less than NUCTotalMax, then program control proceeds to step **445** where the sum of all NUCs for the existing and new flow(s) (NUCTotal) is recomputed, if restrained according to the permitted network policy, if any.

[0072] A further test is performed during step **450** to determine if the sum of all adjusted NUCs for the existing and new flow(s) (adjusted NUCTotal) is less than NUCTotalMax. If it is determined during step **450** that the adjusted NUCTotal is not less than NUCTotalMax, then the new flow is not admitted (step **455**).

[0073] If, however, it is determined during step **450** that the adjusted NUCTotal is less than NUCTotalMax or if it was determined during step **440** that the original NUCTotal is less than NUCTotalMax, then program control proceeds to step **460** where the flow restrictions are adjusted according to the network policy. The flow is admitted during step **470**.

[0074] If the network **150** supports Network Admission Control, then those features are used by the access point **200** to learn about new VoIP calls and for admission control. For example, if the RSVP feature is supported, its mechanisms may be leveraged. For a more detailed discussion of the RSVP feature and admission control, see, for example, Resource Reservation Protocol (RSVP Version 1 Functional Specification), Internet Engineering Task Force (IETF) RFC 2205, 1997; or A Framework for Policy-based Admission Control, IETF RFC 2753, January 2000, each incorporated by reference herein. Standalone Admission Control by the access point **200** is used if the network does not provision admission control features. As "standalone" indicates, the access point 200 performs its operations without any additional information from other devices such as PBXs or SIP Proxies or additional information obtained through protocols, such as RSVP. The access point **200** scans all incoming and outgoing traffic for packets containing call signaling data. If call signaling information is found, the access point **200** operates as if an RSVP message indicating that a new call shall be established were received.

[0075] A special problem in wireless networks is roaming. In this case, the access point **200** that an endpoint is connected to changes during the conversation. This event occurs on Layer **2** and does not trigger any new call signaling. Therefore, if an endpoint roams from one access point **200** to another access point **200**, the ongoing VoIP flow can be detected either by flow detection as described above or by leveraging the inter access point protocol (IAPP) as specified in IEEE 802.11f.

NUCTotalMax

[0076] The NUCTotalMax value, discussed above, may be used to (i) account for certain inaccuracies that might arise if data needed to determine the NUC of flows and auxiliary flows is only partially collected, or (ii) create a network capacity backup that might be needed in some situations (or both).

[0077] Loss and Collisions

[0078] There is a trade-off between the accuracy of the NUC computation and the amount of information that needs to be collected. By keeping track of the traffic and its characteristics, an access point **200** can get a totally accurate picture of the NUCs of flows and auxiliary flows in the wireless network. For example, only one out of ten packets may be used to calculate these values and the NUCs are estimated based on these samples. The resulting inaccuracies can be mitigated by employing a value referred to as NUCTotalMax. In one embodiment, such collisions are accounted without actually measuring them by adjusting the value of NUCTotalMax.

[0079] Due to the nature of the MAC protocol, two stations can attempt to transmit a frame at the same time resulting in a collision. As the data involved in a collision is rendered useless, the NUC of collisions must be accounted for, i.e., the fraction of time per time unit wasted by collisions. It is estimated that if 10 stations simultaneously transmit (a very unlikely scenario), the resulting collisions would waste about 15% of the network capacity. Thus, an exemplary embodiment adjusts the value of NUCTotalMax from 1 to 0.85. With reasonable safety it may be assumed that actual collisions will not exceed the adjusted value of NUCTotalMax and thus that if NUCTotal is less than NUCTotalMax all flows can be transmitted without loss or curtailment due to exceeding the network capacity.

[0080] While this approach is elegant and simple, in most cases the NUC wasted by collisions will be less than 15% and hence some capacity of the network would be wasted. Therefore, a more accurate estimation of the collisions would be beneficial. Extensive simulation studies have shown that the number of collisions is a function of the number of clients associated with an access point **200** and their traffic characteristics. Hence, those values could be tabulated and the NUC of the "wasted" channel capacity can be more accurately estimated by looking up those values.

[0081] Backup Capacity

[0082] While the previous section considered inaccuracies that were deliberately placed into the system by simplifying some part of the data collection process, other inaccuracies result from the bandwidth usage of real-time flows that cannot be eliminated by more detailed data collection. As indicated above, while data flows may be curtailed and still

be useful, VoIP flows need to be given the full required resources. In addition, while the bandwidth of a VoIP flow typically remains fixed during the lifetime of a connection, the NUC of the same connection may change for various reasons.

[0083] For example, the VoIP endpoint in the wireless network may be mobile and roam away from the access point such that the signal strength fades and the transmission speed of the data is decreased from 11 Mbps to 1 Mbps. In one implementation, transmitting a 314 byte frame (a characteristic size for a VoIP frame) in a single sender scenario takes 993 μs at 11 Mbps and 3379 μs at 1 Mbps. Consequently, the NUC of such a stream increases by a factor of approximately 3.4.

[0084] In a network used primarily for data connections, the increase in the NUC of a VoIP flow can be accommodated by further curtailing the NUC of data connections. In a network used primarily for VoIP connections, however, the flow of a VoIP connection cannot be curtailed. Thus, there is a need to have some reserve network capacity to accommodate a change in transmission speed for some of the connections. Since it is unlikely that all of the stations will roam out of range at the same time, this reserve would probably be large enough if it consisted of sufficient NUCs for one or two slow bandwidth connections. However, whether such reserves need to be present at all is a policy question.

[0085] In one exemplary implementation, the NUC values were computed on a per-station basis (i.e., all traffic coming from a particular station of the wireless network is accumulated in a single flow). The system calculates the NUC by leveraging the per-station traffic information as provided by the driver. Apart from this information, the average random back-off value and the average number of collisions were estimated based on the number of active stations in the wireless cell.

Computing the NUC in Special Scenarios

[0086] The present invention could be employed in implementations of the forthcoming IEEE 802.11e standard for QoS in 802.11. The methods applied here also work in different scenarios such as fragmentation.

[0087] RTS/CTS

[0088] Apart from the standard DCF scheme in the IEEE 802.11 standard, the IEEE 802.11 standard also provides a Ready to Send/Clear to Send (RTS/CTS) extension that is particularly useful in wireless networks that might suffer from the hidden station problem. As the standard specifies, the use of the RTS/CTS mechanism is specified on a per-station basis and each station can be configured to use RTS/CTS either always, never or only on frames longer than a specified length. Hence, if the policy of the sending station and the length of the transmitted frame is known, it can be determined whether RTS/CTS is used even without observing the transmission. The policy of each station in the network can either be obtained by observing its behavior or by querying it from the station. On a per-flow basis, the use of RTS/CTS can be estimated by the average size of frames belonging to the flow and the variation of it.

[0089] The overhead added by RTS/CTS consists of two additional SIFSs and the overhead to transmit a 20 byte RTS frame and a 14 byte CTS frame. Similar to the values discussed above, the transmission time for RTS and CTS amount to 129 μs+(20·8)/R and 129 μs+(14·8)/R, respectively, and are thus depending on the transmission rate.

[0090] In contrast to Ethernet, fragmentation in wireless networks is in most cases not the result of a lower maximal frame size of the wireless link (the maximal frame size in 802.11 networks is far higher than in wired Ethernet) but in most cases done deliberately for improving interference stability of the wireless network. The use of fragmentation results in an overhead that is very similar to the one for RTS/CTS.

[0091] IEEE 802.11e

[0092] The forthcoming IEEE 802.11e standard will provide a new MAC scheme, referred to as the Hybrid Coordination Function (HCF), that will combine a centralized polling scheme with a QoS-enhanced version of DCF, referred to as enhanced DCF (EDCF). It is believed that the present invention can be employed in networks conforming to the IEEE 802.11e standard. It is believed that the 802.11e standard will provide a feature that stations can request time-bounded transmission opportunities from the centralized poller which in most cases is identical to the access point **200**. As accurately assessing the channel capacity is very important for determining whether additional traffic can be accommodated, it is believed that the techniques for assessing the channel capacity described herein would be extremely beneficial for 802.11e implementations since the standard defines how to request for such transmission opportunities but does not provide any means to determine the feasibility of such a request.

[0093] It is to be understood that the embodiments and variations shown and described herein are merely illustrative of the principles of this invention and that various modifications may be implemented by those skilled in the art without departing from the scope and spirit of the invention.

We claim:

1. A method for performing admission control in a communication network, said method comprising the step of:

determining whether to accept a new flow based on a network utilization characteristic for said new flow.

2. The method of claim 1, wherein said network utilization characteristic for said new flow is a fraction of time the communication network is busy transmitting data for that flow.

3. The method of claim 1, wherein said determining step further comprises the step of determining a total network utilization characteristic for all flows.

4. The method of claim 3, wherein said determining step further comprises the step of determining whether said total network utilization characteristic for all flows is below an upper limit.

5. The method of claim 4, wherein said upper limit provides a tolerance for backup capacity.

6. The method of claim 4, wherein said upper limit provides a tolerance for inaccurate data.

7. The method of claim 4, further comprising the step of curtailing one or more existing flows if said total network utilization characteristic for all flows is not below said defined upper limit.

8. The method of claim 1, further comprising the step of classifying said new flow as a voice call.

9. The method of claim 1, further comprising the step of classifying said new flow as a data call.

10. The method of claim 1, wherein said network utilization characteristic is based on a transmission time of a frame.

11. The method of claim 10, wherein said transmission time of a frame includes a time to transmit a data frame.

12. The method of claim 10, wherein said transmission time of a frame includes one or more inter-frame spacing intervals.

13. The method of claim 10, wherein said transmission time of a frame includes one or more acknowledgement intervals.

14. The method of claim 10, wherein said transmission time of a frame includes one or more back-off intervals.

15. A system for performing admission control in a communication network, comprising:

a memory that stores computer-readable code; and

a processor operatively coupled to said memory, said processor configured to implement said computer-readable code, said computer-readable code configured to:

determine whether to accept a new flow based on a network utilization characteristic for said new flow.

16. The system of claim 15, wherein said network utilization characteristic for said new flow is a fraction of time the communication network is busy transmitting data for that flow..

17. The system of claim 15, wherein said wherein said processor is further configured to determine a total network utilization characteristic for all flows.

18. The system of claim 17, wherein said wherein said processor is further configured to determine whether said total network utilization characteristic for all flows is below an upper limit.

19. The system of claim 18, wherein said upper limit provides a tolerance for backup capacity.

20. The system of claim 18, wherein said upper limit provides a tolerance for inaccurate data.

21. The system of claim 18, wherein said processor is further configured to curtail one or more existing flows if said total network utilization characteristic for all flows is not below said defined upper limit.

22. An article of manufacture for performing admission control in a communication network, comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

a step to determine whether to accept a new flow based on a network utilization characteristic for said new flow.

* * * * *

US006697364B1

(12) **United States Patent**
Kekki et al.

(10) **Patent No.: US 6,697,364 B1**
(45) **Date of Patent: Feb. 24, 2004**

(54) **STATISTICAL MULTIPLEXING IN A TELECOMMUNICATIONS NETWORK**

(75) Inventors: **Sami Kekki**, Helsinki (FI); **Jyri Suvanen**, Helsinki (FI)

(73) Assignee: **Nokia Corporation**, Espoo (FI)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 435 days.

(21) Appl. No.: **09/616,313**

(22) Filed: **Jul. 17, 2000**

Related U.S. Application Data

(63) Continuation of application No. PCT/FI99/00053, filed on Jan. 28, 1999.

(30) **Foreign Application Priority Data**

Jan. 28, 1998 (FI) 980189

(51) **Int. Cl.**[7] **H04J 3/02**; H04J 3/16

(52) **U.S. Cl.** **370/389**; 370/471; 370/474; 370/535; 714/749

(58) **Field of Search** 370/389, 395.01, 370/431, 433, 465, 468, 470, 471, 474, 535; 714/746, 748, 749

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,082,922	A	* 4/1978	Chu	179/15 BA
4,494,232	A	* 1/1985	Dambrackas et al.	370/80
4,700,341	A	* 10/1987	Huang	370/80
5,153,876	A	* 10/1992	Sin	370/85.1
5,675,642	A	* 10/1997	Sone	370/389
5,699,356	A	12/1997	Beever et al.	370/329
5,768,265	A	* 6/1998	Toyohara	370/282
6,229,821	B1	* 5/2001	Bharucha et al.	370/471

FOREIGN PATENT DOCUMENTS

EP	0407 367 A3	1/1991
WO	WO 96/35299	11/1996
WO	WO 97/48200	12/1997

OTHER PUBLICATIONS

International Search Report for PCT/FI99/00053.

M. Mouly & M. Pautet, "The GSM System for Mobile Communications", Palaiseau, France, 1992, ISBN:2-9507190-0-7.

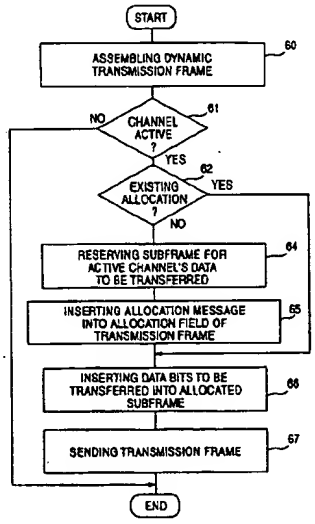
* cited by examiner

Primary Examiner—Alpus H. Hsu
(74) *Attorney, Agent, or Firm*—Squire, Sanders & Dempsey L.L.P.

(57) **ABSTRACT**

A method carries out statistical multiplexing in a telecommunications network comprising a transmitting unit for transferring channels containing information from at least two traffic sources, a receiving unit and a transmission link between them. The information to be transmitted in the telecommunication system is transferred over the transmission link in transmission frames. The method is characterized by assembling a variable-length transmission frame comprising an allocation field and an information field, identifying an active channel, allocating an information field transmission block from the transmission frame to the active channel's information to be transferred, inserting the information to be transferred into the allocated transmission block, adding the channel allocation information into the allocation field of the transmission frame in which the channel's information is for the first time continuously transferred, maintaining the channel's allocation for as long as the channel is continuously active, and transmitting the transmission frame to the receiving unit.

20 Claims, 10 Drawing Sheets



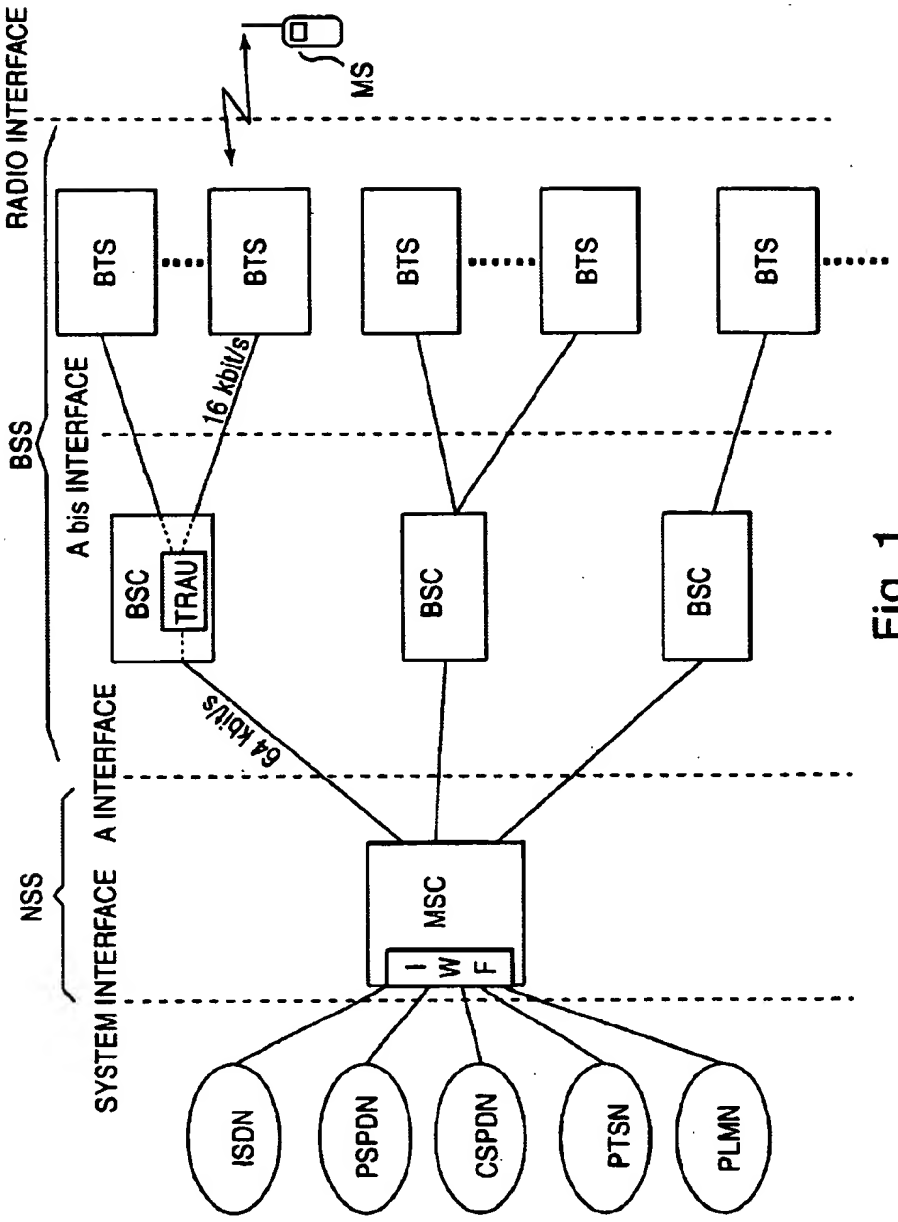

RADIO INTERFACE
MS
BTS
BTS
BTS
BTS
BTS
BSS
A bis INTERFACE
16 kbit/s
BSC
TRAU
BSC
BSC
64 kbit/s
A INTERFACE
NSS
SYSTEM INTERFACE
MSC
I W F
ISDN
PSPDN
CSPDN
PTSN
PLMN

Fig. 1

octet number	bit number 1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	1	C1	C2	C3	C4	C5	C6	C7
3	C8	C9	C10	C11	C12	C13	C14	C15
4	1	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X
6	1	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X
8	1	X	X	X	X	X	X	X
9	X	X	X	X	X	X	X	X
10	1	X	X	X	X	X	X	X
11	X	X	X	X	X	X	X	X
12	1	X	X	X	X	X	X	X
13	X	X	X	X	X	X	X	X
14	1	X	X	X	X	X	X	X
15	X	X	X	X	X	X	X	X
16	1	X	X	X	X	X	X	X
17	X	X	X	X	X	X	X	X
18	1	X	X	X	X	X	X	X
19	X	X	X	X	X	X	X	X
20	1	X	X	X	X	X	X	X
21	X	X	X	X	X	X	X	X
22	1	X	X	X	X	X	X	X
23	X	X	X	X	X	X	X	X
24	1	X	X	X	X	X	X	X
25	X	X	X	X	X	X	X	X
26	1	X	X	X	X	X	X	X
27	X	X	X	X	X	X	X	X
28	1	X	X	X	X	X	X	X
29	X	X	X	X	X	X	X	X
30	1	X	X	X	X	X	X	X
31	X	X	X	X	X	X	X	X
32	1	X	X	X	X	X	X	X
33	X	X	X	X	X	X	X	X
34	1	X	X	X	X	X	X	X
35	X	X	X	X	X	X	X	X
36	1	X	X	X	X	X	X	X
37	X	X	X	X	X	X	X	X
38	1	X	X	X	X	X	C16	C17
39	C18	C19	C20	C21	T1	T2	T3	T4

Fig. 2

octet number	bit number 1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	1	C1	C2	C3	C4	C5	C6	C7
3	C8	C9	C10	C11	C12	C13	C14	C15
4	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1
29	1	1	1	1	1	1	1	1
30	1	1	1	1	1	1	1	1
31	1	1	1	1	1	1	1	1
32	1	1	1	1	1	1	1	1
33	1	1	1	1	1	1	1	1
34	1	1	1	1	1	1	1	1
35	1	1	1	1	1	1	1	1
36	1	1	1	1	1	1	1	1
37	1	1	1	1	1	1	1	1
38	1	1	1	1	1	1	C16	C17
39	C18	C19	C20	C21	T1	T2	T3	T4

Fig. 3

Fig. 4

octet number	bit number 1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
1	CH1	CH1	CH2	CH2	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X
9	X	X	X	X	X	X	X	X
10	X	X	X	X	X	X	X	X
•								
•								
•								
29	X	X	X	X	X	X	X	X
30	X	X	X	X	X	X	X	X
31	X	X	X	X	X	X	X	X

Fig. 5

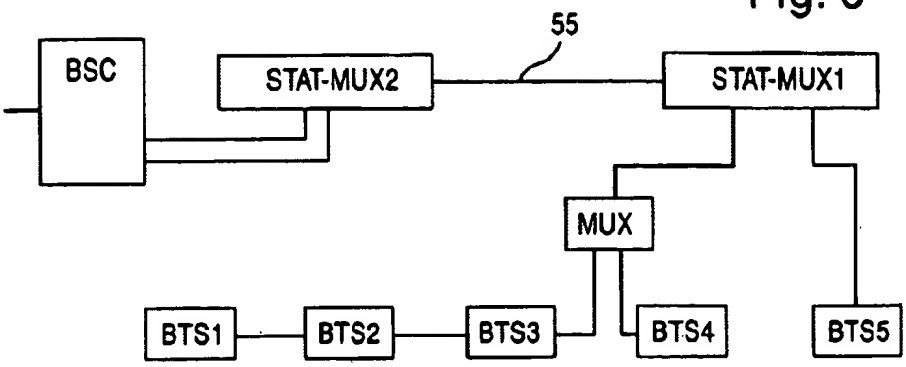


Fig. 7

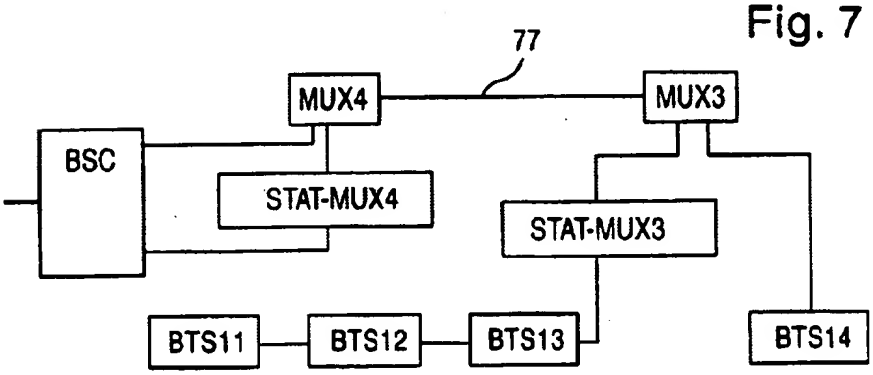


Fig. 9

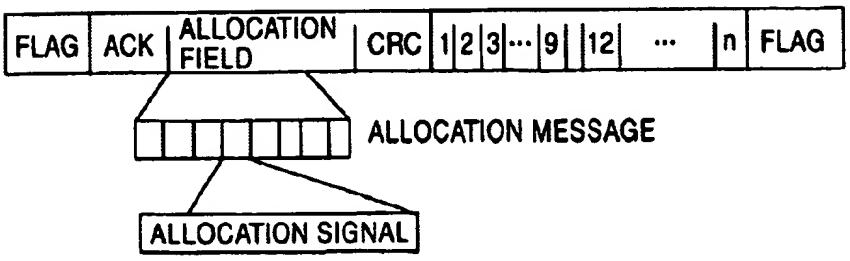


Fig. 6

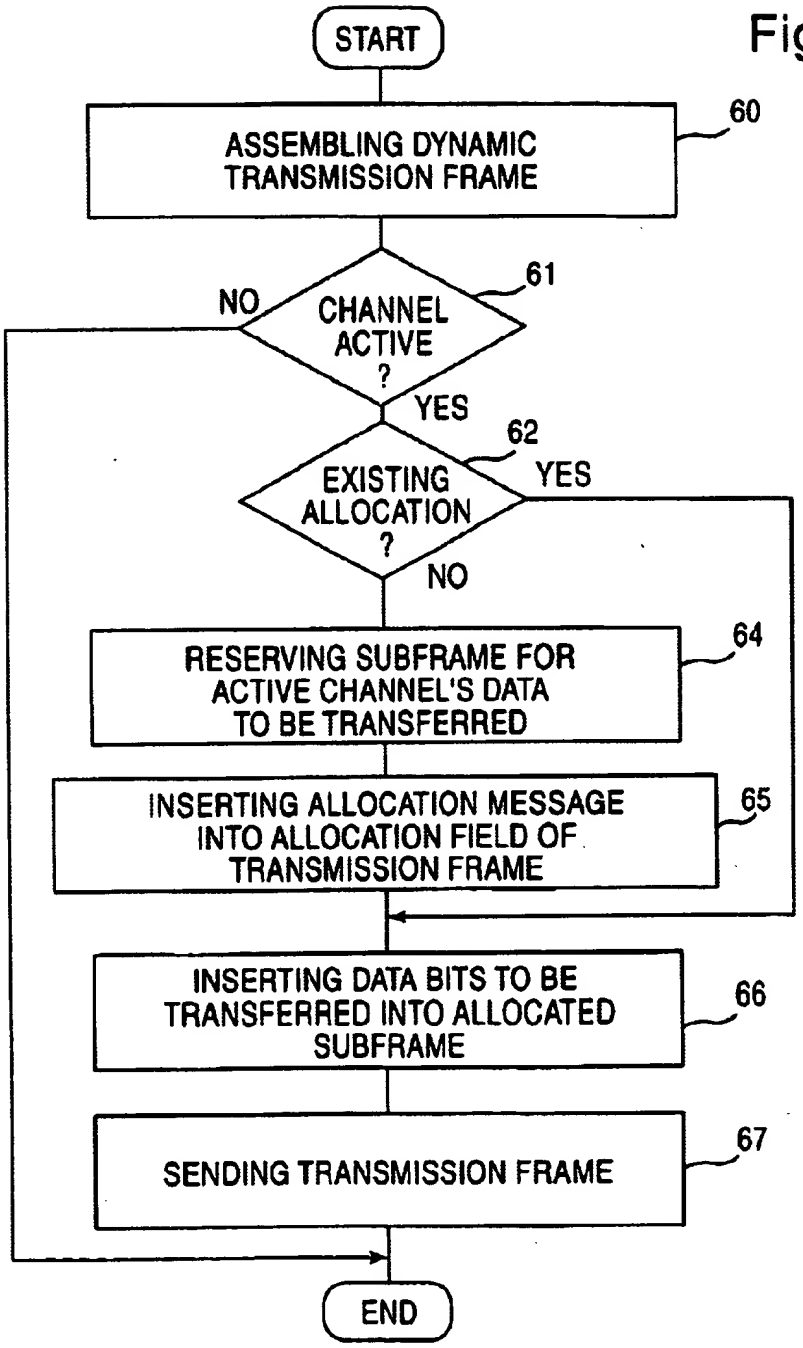

START
ASSEMBLING DYNAMIC TRANSMISSION FRAME
60
CHANNEL ACTIVE ?
61
NO
YES
EXISTING ALLOCATION ?
62
YES
NO
RESERVING SUBFRAME FOR ACTIVE CHANNEL'S DATA TO BE TRANSFERRED
64
INSERTING ALLOCATION MESSAGE INTO ALLOCATION FIELD OF TRANSMISSION FRAME
65
INSERTING DATA BITS TO BE TRANSFERRED INTO ALLOCATED SUBFRAME
66
SENDING TRANSMISSION FRAME
67
END

Fig. 8

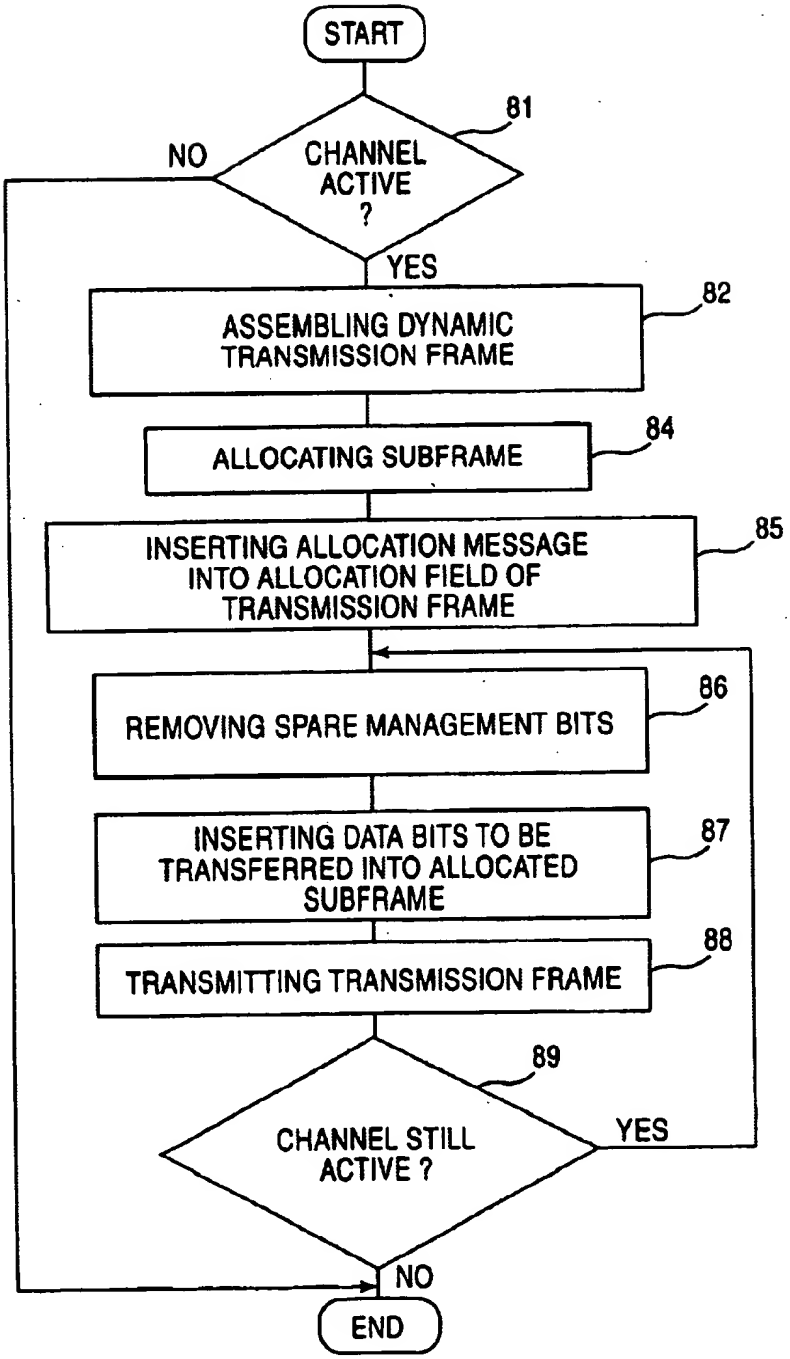

START
81
CHANNEL ACTIVE ?
NO
YES
82
ASSEMBLING DYNAMIC TRANSMISSION FRAME
84
ALLOCATING SUBFRAME
85
INSERTING ALLOCATION MESSAGE INTO ALLOCATION FIELD OF TRANSMISSION FRAME
86
REMOVING SPARE MANAGEMENT BITS
87
INSERTING DATA BITS TO BE TRANSFERRED INTO ALLOCATED SUBFRAME
88
TRANSMITTING TRANSMISSION FRAME
89
CHANNEL STILL ACTIVE ?
YES
NO
END

Fig. 10

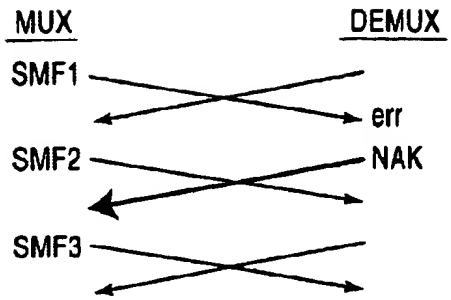

MUX
DEMUX
SMF1
err
SMF2
NAK
SMF3

Fig. 12

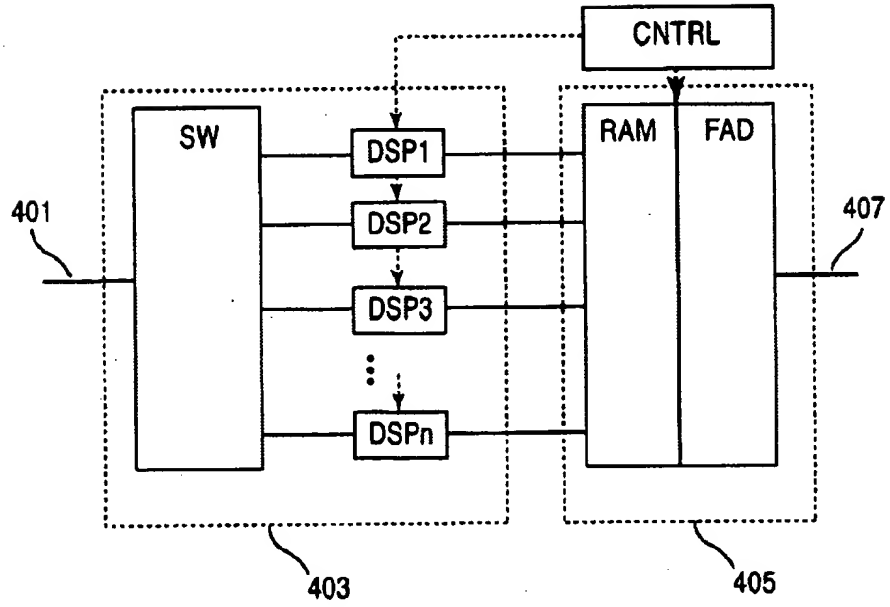

CNTRL
SW
DSP1
DSP2
DSP3
DSPn
RAM
FAD
401
407
403
405

Fig. 11

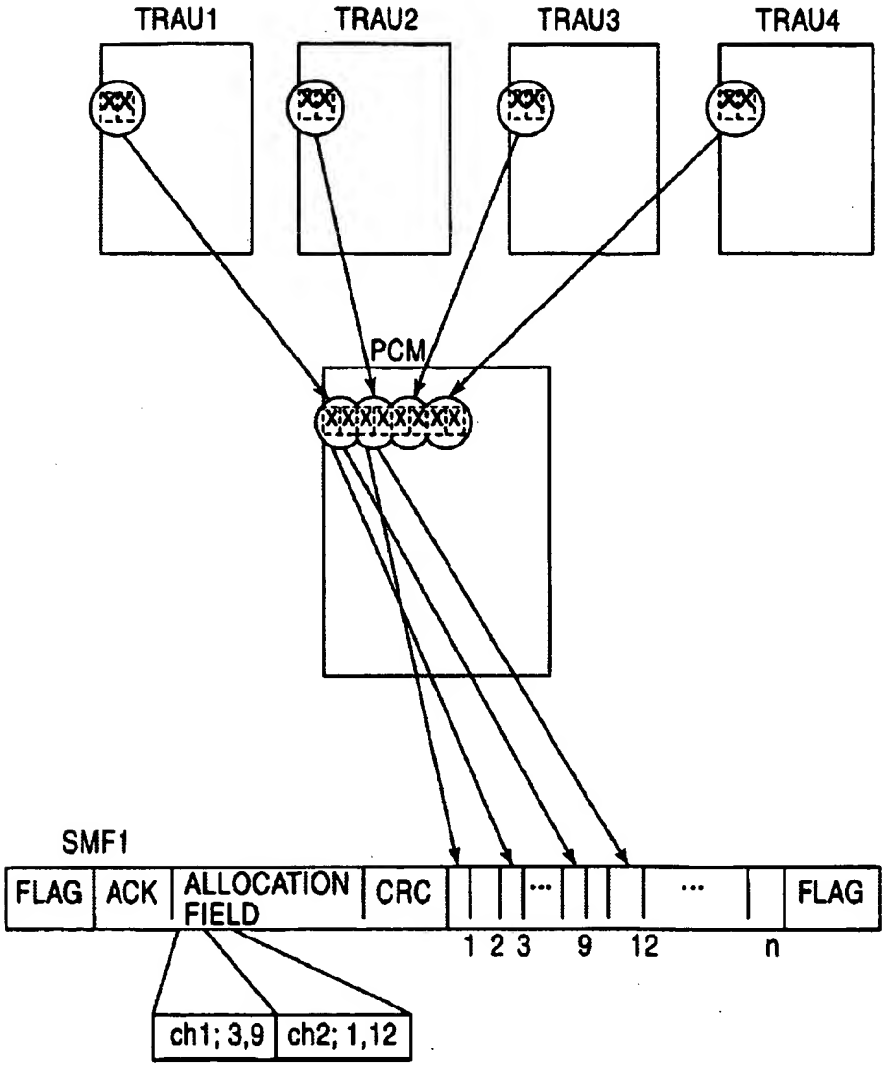
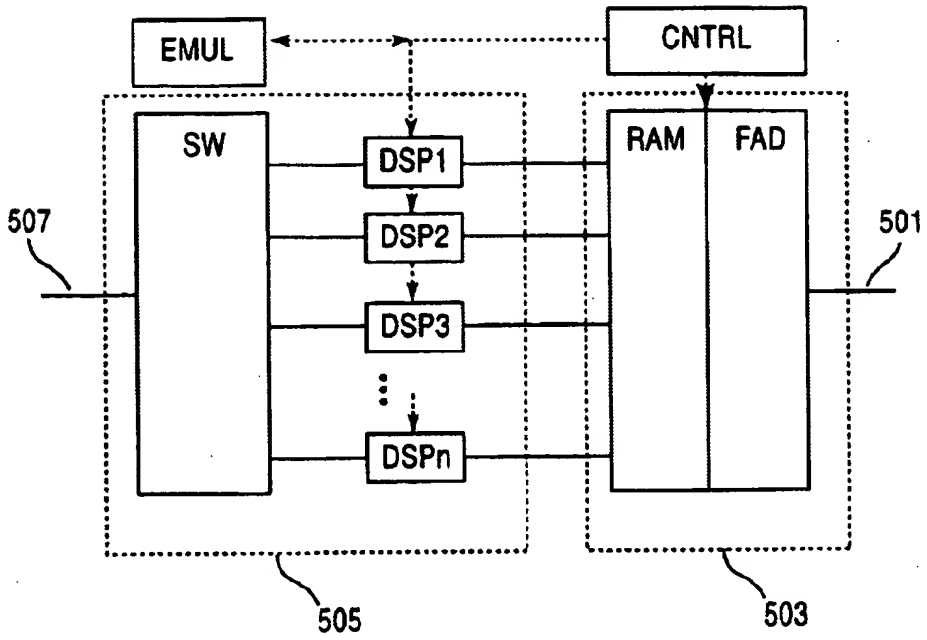

TRAU1
TRAU2
TRAU3
TRAU4
PCM
SMF1
FLAG
ACK
ALLOCATION FIELD
CRC
FLAG
1 2 3 9 12 n
ch1; 3,9
ch2; 1,12

Fig. 13



STATISTICAL MULTIPLEXING IN A TELECOMMUNICATIONS NETWORK

This application is a continuation of PCT/FI99/00053 filed Jan. 28, 1999.

FIELD OF THE INVENTION

The invention relates to statistical multiplexing in a telecommunications network, as well as a statistical multiplexer, and statistical demultiplexer.

TECHNICAL BACKGROUND

FIG. 1 in the accompanying drawings shows a simplified block diagram of GSM mobile communication system (GSM. Global System for Mobile Telecommunications). The network subsystem (NSS) comprises a mobile services switching centre MSC which communicates with other mobile services switching centres, and either directly or via the system interface of a gateway mobile services switching centre (GMSC), the mobile communication system is connected to other networks, such as the public switched telephone network (PSTN), an integrated services digital network (ISDN), other mobile communication networks such as the public land mobile network (PLMN) and packet-swztched public data networks (PSPDN) and circuitswitched public data networks (CSPDN). The mobile services swishing centre comprises network interworking functions (IWF) by means of which the GSM network can be adapted to other networks. The network subsystem NSS is connected via the A interface to the base station subsystem (BSS) which comprises base station controllers BSC, each controlling base stations BTS that are connected to them. The interface between the BSC and base stations BTS connected thereto is the A bis interface.Base stations BTS for their part communicate on the radio path with mobile stations MS over the radio interface. The operation of the entire system is monitored by an Operation and Maintenance Centre, OMC.

The transcoder/rate adaptor unit (TRAU) is a part of the base station subsystem BSS and may be located at the base station controller BSC, as shown in FIG. **1**, or alternatively at the mobile services switching centre MSC. The transcoders convert speech from a digital format to another, for example 64 kbit/s PCM received over the A interface from the MSC, into data to be transmitted to the base station, and vice versa. One 64 kbit/s PCM channel carries four speech/data connections, which means that the rate of one speech/data channel on this link is 16 kbit/s.

The mobile station MS transmits user data over the radio interface on the radio channel at 13 kbit/s or 5.6 kbit/s, as specified in the standard. The base station BTS receives the data of the traffic channel and transfers it to the 64 kbit/s timeslot of the PCM circuit In addition, the three other traffic channels of the same carrier are inserted into the same timeslot, i.e. channel, resulting in that the transfer rate per connection is 16 kbit/s. At the BSC, the TRAU converts the coded 16 kbit/s digital information into the 64 kbit/s channel, and on this channel the data is transferred into the IWF unit at the MSC. The IWF carries out the necessary modulation and rate adaptation, after which the data is transmitted to some other network. Thus, user data is transferred over fixed connections in the uplink direction from the BTS to the BSC and the MSC, and correspondingly, the data to be transmitted to the MS is transferred in the downlink direction from the MSC via the BSC to the BTS and from thereon over the radio interface to the MS. The

channel codec unit (CCU) of the base station carries out the conversion of the signal received on the radio channel into the channel of the PCM time slot in the trunk circuit over the A bis interface, and the conversion of the signal received over the A bis interface into the form transmitted to the radio path. The TRAU carries out the conversion operations for the signals to be transferred over the A interface.

The user data is transferred over the A bis interface from the BTS to the TRAU in a fixed-length TRAU frame. FIG. **2** shows in bit diagram form a TRAU data frame used to transfer a signal at the data rate of 13 kbit/s. The TRAU frame comprises 40 octets numbered **0**, . . . , **39**, its total length thus being 320 bits and duration 20 ms. Synchronization between the unit that transmits the TRAU frame and the one that receives it is achieved with synchronization bits that are shown in FIG. **2** as 0 bits and 1 bits. The 0 bits in the first two octets of the TRAU speech frame are used for carrying out the actual synchronization, and the 1 bits in the first bit position in the other even octets except the second and fourth, are used to ensure that elsewhere in the data frame there are no two-octet-long sequences of successive 0 bits that would look like a synchronization sequence. One TRAU speech frame contains 35 synchronization bits per the total number of 320 bits. The TRAU frame of FIG. **2** shows control bits C1–C21, timing bits T1–T4 and the user data bits denoted with X. Unused data bits are set to 1-state, for example, for the duration of breaks in the data transmission, whereby the TRAU frame to be transferred is in accordance with the idle speech frame of FIG. **3**. The BFI control bit (Bad Frame Indicator) in the idle frame is used to indicate that the frame does not contain speech information.

The bits in the TRAU frames are normally transferred over the PCM line in a PCM frame according to ITU-T Recommendation G.704, whose structure is shown in FIG. **4**. The frame comprises 32 octets numbered **0**, . . . , **31**. The frame duration with the transfer rate of 2 Mbit/s is 125 μs. The 0-bits of the first octet are the frame synchronization. CH1, CH2 and X-bits marked in FIG. **4** comprise the bits to be transferred from different TRAU frames, so that at the transfer rate of 8 kbit/s the bit to be transferred from each TRAU frame i.e. channel is inserted into one bit position in the PCM frame, and at the transfer rate of 16 kbit/s user data bits of the TRAU frame are inserted into two bit positions in the PCM frame. FIG. **4** shows a PCM frame with the transfer rate of 16 kbit/s as regards the transfer of channels CH1 and CH2.

In the GSM system, the transmission of bit streams from different sources is enhanced e.g. by means of multiplexing and concentrator equipment. The bit steams are interleaved to e.g. 2 Mbit/s buses with TDM (Time Division Multiplex) so that each channel in the system is allocated a dedicated timeslot, which the channel always uses.

Discontinuous transmission (DTX) refers to a method in which the mobile station's transmission to the radio path may be interrupted for the duration of pauses in speech. The aim is to reduce the power consumption of the mobile station, a very significant issue for it. In the GSM system, for example, the speech activity of the signal transmitted from the mobile station to the bas station is monitored at the mobile station, and if no speech information exists, the mobile station's transmission to the radio path is cut. When the mobile station's MS transmission has been discontinued, the base station BTS generates idle frames according to FIG. **3** and transmits them forward to the base station controller BSC. When speech again begins, it is coded at the mobile station and transmitted to the base station in the correct timeslot. In order for the receiving party not to sense the

transmission cut as unpleasant total silence, comfort noise parameters (SID, Silence Descriptor) are transmitted to the base station at specific intervals, 0.5 s in the GSM system, which are used to generate comfort noise simulating background noise in the speech decoder.

The contents of the user bits in the TRAU frame are indicated with the aid of control bits. In the uplink direction, control bits C13 and C14 in the BFI, SID and TAF (Time Alignment Flag) flags in the TRAU frame are used to indicate whether the TRAU frame to be transferred contains speech or SID comfort noise parameters. In the idle TRAU frame according to FIG. 3, these control bits are employed to indicate that the frame does not contain payload. Correspondingly in the downlink direction, control bit C16 in the TRAU frame according to FIG. 2 is used to indicate whether the TRAU frame to be transferred contains speech (SP=1) or something else than speech information (SP=0).

The problem with the prior art speech transfer described above is that useless information is transferred in it, for example idle TRAU speech frames during pauses in speech, which leads to transfer capacity being wasted. Due to the fixed channel allocation of TDM (Time Division Multiplexing), the channel is continuously reserved for use by one traffic source regardless of the actual demand for transfer capacity of said traffic source. Because in speech traffic the subscribers as a general rule speak alternately, and because the speaking party will have irregular pauses when talking, the efficiency of the reserved channel is roughly less than 50% of the duration of the call. Particularly when connecting the traffic of several base stations to the same bus the average efficiency of the bus is a mere 30% of the theoretical maximum utilization degree because calls are often set up at different times, which means that only part of the channels on the bus are used simultaneously. With the prior art transfer, additional problems are caused by redundancy in the transfer of synchronization bits, caused by channel-specific synchronization of the TRAU frame structure when 35 synchronization bits are transferred in each TRAU frame.

SUMMARY OF THE INVENTION

The aim of this invention is to enhance data transfer so that more payload can be transferred with low transfer capacity than previously.

The object is achieved with the inventive method, characterized by that which is claimed in the independent claim 1. The preferred embodiments of the invention are disclosed in the dependent claims 2–11.

The invention additionally relates to a statistical multiplexer, a statistical demultiplexer and a transmission frame which are characterized by that which is claimed in the independent claims 12, and 16. The preferred embodiments of the invention are disclosed in the dependent claims.

The invention is based on the idea that statistical multiplexing is employed to assign bus capacity only for the active channels. In this process, the bits of the active channels are placed into the transmission blocks of a variable-length transmission frame and sufficient identification data is added to the frame. The multiplexer sends this inventive transmission frame to the transmission bus whose capacity is advantageously in the exclusive use of the frame in question. On the transmission link, the frame is routed in the network nodes to the correct terminal node, in which the bits in the frame are demultiplexed to their respective channels on the basis of the identification data. In other words, the inventive statistical multiplexing transfers the bits from a prior art regular frame structure to the inventive variable-length transmission frame for the duration of the transfer, and back to the regular transmission frame at the receiving end. Additionally, information transfer is further enhanced in the particular embodiments of the invention, by minimizing the number of bits to be transferred by e.g. removing unnecessary synchronization bits and/or control bits, and possibly by compressing the assembled transmission frame prior to transfer.

Such statistical multiplexing provides the advantage that channel allocation is extremely dynamic, resulting in lower data transfer capacity and, secondly, the assigned capacity being brought into more efficient use. By assigning data transfer capacity for use by data transfer of the active channels, only, temporally-interleaved signals can be inserted in the same transmission block. With the inventive statistical multiplexing, the operator is able to add capacity to its network without further investments in transmission lines when e.g. more TRXs can be connected to one 2 Mbit/s transmission line than previously.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will below be described in connection with the preferred embodiments, with reference to examples in FIGS. 5–13 in the accompanying drawings, in which.

FIG. 1 shows the essential parts of a mobile communications network from the invention's point of view;

FIG. 2 shows the structure of a TRAU frame;

FIG. 3 shows the structure of an idle TRAU frame;

FIG. 4 shows the structure of a PCM frame;

FIG. 5 shows a first exemplary structure of the inventive statistical multiplexing arrangement as a block diagram;

FIG. 6 shows the first embodiment of the inventive statistical multiplexing as a flow chart;

FIG. 7 shows a second exemplary structure of the inventive statistical multiplexing arrangement as a block diagram;

FIG. 8 shows the fourth embodiment of the inventive statistical multiplexing as a flow chart;

FIG. 9 shows the structure of the inventive transmission frame;

FIG. 10 shows an exemplary case of how a transmission error in an allocation message affects the transfer of SMF frames;

FIG. 11 shows an exemplary case of assembling the inventive transmission frame;

FIG. 12 shows an embodiment of the inventive statistical multiplexer as a block diagram; and,

FIG. 13 shows an embodiment of the inventive statistical demultiplexer as a block diagram.

DETAILED DESCRIPTION OF THE INVENTION

The present invention can be applied in connection with any communications network. Below, the invention is described by way of example mainly in connection with the digital GSM mobile communications system. FIG. 1 shows the simplified structure of the GSM network, described above. For a more detailed description of the GSM system, reference is made to the GSM Recommendations and "The GSM System for Mobile Communications", M: Mouly & M. Pautet, Palaiseau, France, 1992, ISBN:2-9507190-0-7.

A transmission frame assembled according to the present invention will in this application be referred to as an SMF

frame (Statistical Multiplexing Frame). The channel-specific data block carried in the SMF frame is termed a subframe. The structure of the SMF frame will be described in greater detail below.

FIG. **5** shows the structure of an inventive statistical multiplexing arrangement as a block diagram. The inventive statistical multiplexes STAT-MUXn illustrated in the figures of this application comprise both a statistical multiplexer for carrying out mutiplexing and a demultiplexer for disassembling the multiplexing. In accordance with prior art, FIG. **5** shows the traffic of base stations BTS1–BTS3 being combined into a common bus and the traffic of this bus being concentrated with the traffic of base station BTS in a prior art multiplexer MUX. According to the present invention, the traffic of base station BTS5 and the concentrated traffic from multiplexer MUX are applied to the inventive multiplexer STAT-MUX1, which performs the inventive statistical multiplexing for the traffic and transmits the bits in an inventive SMF frame to the inventive statistical demultiplexer STAT-MUX2 over transmission link **55**. In the GSM system, transmission link **55** travels across the A bis interface. The statistical demultiplexer STAT-MUX2 at the receiving end demultiplexes the information carried in the SMF frame into a prior art fixed-length frame, e.g. into a PCM frame that is forwarded to the base station controller BSC. Correspondingly, the traffic sent from the BSC to the base stations is statistically demultiplexed according to the invention in statistical demultiplexer STAT-MUX2 which transmits the bits to statistical demultiplexer STAT-MUX1 in an SMF frame over transmission link **55**. In STAT-MUX1, the information in the transmission frame is demultiplexed into a prior art fixed-length frame, such as a PCM frame. This prior art frame is passed on, according to prior art, to the base stations. Transmission link **55** and the base station connections of statistical multiplexer STAT-MUX1 are 2 Mbit/s transmission lines, for example.

In the following, the invention will be described in closer detail in the light of its preferred embodiment with reference to FIG. **6**. In the preferred embodiment of the invention, the radio link uses discontinuous transmission (DTX) in the uplink and/or downlink direction. FIG. **6** shows the inventive method for one active channel. At step **60** in FIG. **6**, the inventive dynamic variable-length SMF frame is formed In a statistical multiplexer, in the case of FIG. **6** e.g. in multiplexer STA-MUX1. Based on the information of the channel which was received by the statistical multiplexer and which is meant to be transferred over the A bis interface, it is monitored at step **61** whether the channel in question is active, e.g. on the basis of the information contents indicated by the control bits in the TRAU frame. The channels that carry speech information and, secondly, SID comfort noise parameters are interpreted as active channels, but not the channels that carry idle speech frames, such as TRAU idle speech frames according to FIG. **3**. After identifying an active channel, it is checked at step **62** whether the channel already has an existing allocation in the SMF frame. The allocations are not actually removed for the duration of speech pauses, but the channel may have been assigned for use by another active channel during the pause. If the channel does not have an existing allocation, a subframe is allocated from the SMF frame formed at step **64**, i.e. a data transmission block, for transfer of the active channel's information. The information of the channel in question has not been transferred in the previous transmitted SMF frame, but the allocation for the information is new. In the case of this new allocation, an allocation message is inserted into the allocation field of the SMF frame at step **65** of FIG. **6**. In the allocation message is indicated the identification data of the subframe used for data transmission of the channel in question. At step **66**, the active channel bits are inserted into the allocated subframe. Correspondingly, bits of other active channels are inserted into the other subframes of the SMF frame. If the channel has a previous existing allocation for the SMF frame, a move is made from step **62** directly to step **66**. In the case of a previous allocation, for example, if uninterrupted activity on the channel continues, the bits in turn for transfer are inserted into the same subframe allocated from the SMF frame as in the previous SMF frame. In such a case, no allocation message is transmitted in the allocation field of the SMF frame as regards the allocation of this channel, because the receiving end knows about the allocation or the basis of the allocation message sent in the previous SMF frame. At step **67** in FIG. **6**, the SMF frame thus formed is transmitted to the transmission line, to transmission line **55** in the case of FIG. **5**. For the next SMF frame, the functionality of the FIG. **6** flow-chart begins from the top by forming a new SMF frame at step

The insertion of the active channel's information into the subframes reserved for this purpose in each SMF frame is continued until the channel is no longer active, for example when the channel carries an idle speech frame that is not forwarded according to the invention. When the channel reactivates, information transmission may be continued in the same subframe of the SMF frame in case the subframe has not been allocated to another active channel during the transmission pause. In such a case, too, a move is made from step **62** in FIG. **6** directly to step **66**. During the pauses in speech, idle TRAU frames that were not transmitted are, if need be, generated in the receiving unit in a way to be described more closely below.

FIG. **7** shows another network arrangement to implement the inventive statistical multiplexing. In the exemplary case of FIG. **7**, statistical multiplexer STAT-MUX3 inserts bit streams from three base stations BTS11–BTS13 connected to the same transmission bus into an inventive SMF frame. Prior art multiplexer MUX3 concentrates traffic received from base station BTS14 and statistical multiplexer STAT-MUX3 according to prior art, and transfers data thus multiplexed over transmission link **77** to prior art demultiplexer MUX4. After carrying out demultiplexing, MUX4 transfers the inventive SMF frame to statistical demultiplexer STAT-MUX4 and the other frames directly to the base station controller BSC. Statistical demultiplexer STAT-MUX4 inserts the transmitted bits from the SMF frame to a prior art fixed-length frame, such as a PCM frame, and forwards this frame to the BSC. Correspondingly, information from base station-controller BSC to base stations BTS11–BTS13 is transferred statistically multiplexed by means of statistical multiplexer STAT-MUX4 in an SMF frame via multiplexer MUX, transmission link **77**, and demultiplexer MUX3 to statistical multiplexer STAT-MUX3, which inserts the bits of the SMF frame to a prior art frame for forwarding to base stations BTS11–BTS13. Transmission link **77** Is, e.g., a 2 Mbit/s transmission line, and the connection between STAT-MUX3 and MUX3 is, e.g., an n*64 kbit/s transmission line.

In the second embodiment of the invention, a plurality of base stations BTS are connected to the statistical multiplexer, as is the case with base stations BTS11–BTS13 in FIG. **7**. These base station connections are advantageously low-capacity transmission links. As calls are often set up at different times at different base stations, the inverted statistical multiplexing can be utilized by transmitting the calls of the base stations in the inventive SMF frame on a transmission bus that has lower data transfer capacity

than that which would be required by simultaneous traffic via the base stations. In the following, the second embodiment of the invention will be described in more detail still referring to FIG. **6**. In the second embodiment of the invention, the activity study of step **61** in FIG. **6** is carried out by monitoring ongoing calls, i.e. traffic activity, of the traffic transmitted by each base station, base stations BTS11–BTS13 in the case of FIG. **7**. Each ongoing call is understood to constitute a channel of its own, whose transfer is continued as shown in FIG. **6** until the call is terminated. In other respects in the second embodiment of the invention, the SMF frame is formed the subframe is allocated, and the allocation message is transmitted in the first SMF frame as described in connection with the preferred embodiment of the invention.

In the preferred and second embodiments of the invention, the checking according to step **62** can be bypassed and a new subframe be always allocated for a new period of activity. In such a case, however, the capacity saving will be lost because the data transfer efficiency is greatly reduced as large part of the transfer capacity has to be allocated for transfer of allocation information.

The third embodiment of the invention combines the functionality of the preferred and second embodiments described above. This means that the third embodiment employs discontinuous transmission (DTX) in the uplink and/or downlink direction and the traffic of a plurality of base station is connected to the statistical multiplexer. In the third embodiment of the invention, then, the activity monitoring scheme at step **61** of FIG. **6** first involves monitoring the channel's activity according to the second embodiment of the invention, and additionally the speech activity of the channel with traffic activity according to the preferred embodiment of the invention. In the third embodiment, the discontinuous transmission function may be in use only on some channels, whereby speech pause monitoring according to the invention is carried out e.g. only for this channel, and, as to the other channels, only traffic activity is monitored.

FIG. **8** shows the fourth embodiment of the inventive statistical multiplexing as a flow chart. In the fourth embodiment, extra management bits are removed from the bit stream to be transferred prior to inserting the bits into the inventive SMF frame. Extra management bits in this application refer to control bits and bits in the synchronization pattern, the unnecessary bit stream formed by which can be removed from among bits to be transferred. At step **81** of FIG. **8** it Is monitored whether the channel is active. This is done based on e.g. SID/speech activity and/or traffic activity, as described above in connection with the preferred, second and third embodiments of the invention. According to the invention, a dynamic SMF transmission frame is formed (step **82**), from which a subframe is allocated to the active channels' bits to be transferred (step **84**). The allocation message is placed into the allocation field of the assembled transmission frame (step **85**) as described above in connection with the preferred embodiment of the invention. In the fourth embodiment, extra management bits are removed from the active channels' bits to be transferred, e.g. at least part of the synchronization bits and/or at least part of the control bits. In such a case, all or part of the synchronization bits in the TRAU frame of FIG. **2** are removed, whereby of the bits in the TRAU frame just the bits in the bit stream that remain after the bit removal are inserted into the inventive SMF frame (step **87** in FIG. **8**). The transmission frame formed at step **88** of FIG. **8** is transmitted **10** the receiving party over the transmission link. The activity of the channel is checked at step **89**, and the treatment according to steps **86–89** is repeated for the active channel. In the fourth embodiment of the invention, an existing allocation can be checked, too, as described in connection with the preferred embodiment with reference to step **62** in FIG. **6**. The total or partial removal of synchronization bits and/or control bits is indicated to the receiving end, e.g. as an initial setting or within the allocation message, whereby the receiving statistical demultiplexer is able to generate the needed synchronization and/or control bits to the received bit stream. As a TRAU frame is known to begin from the beginning of the subframe of the SMF frame containing the channel allocation message, the synchronization bits can be added to the bit stream at the receiving end. Removal of all synchronization bits contributes to saving 35 bits per TRAU frame of the transfer capacity of the bus, i.e. approximately 11% of the transfer capacity. By removing spare C-bits, further capacity saving will be achieved. The synchronization of the data to be transferred by means of the SMF frame can, for the duration of the transfer, be replaced by a smaller synchronization pattern, which the statistical demultiplexer at the receiving end removes and generates in its stead synchronization bits required for transfer of a regular frame structure.

In the fifth embodiment of the invention, the formed SMF frame is compressed prior to sending it to the transmission bus. By means of compression, the data is further reduced, by e.g. compression according to recommendation V.42bis. Compression is applicable for use in connection with any of the embodiments described above.

FIG. **9** shows an inventive transmission frame which is formed by a statistical multiplexer out of data on channels received e.g. in the form of a PCM frame. The formed SMF transmission frame is transmitted to the receiving end's statistical demultiplexer, which demultiplexes the frame structure used during the transfer an reinserts the data into a prior art regular frame, such as a PCM frame. The transfer blocks denoted with numbers **1, 2**, . . . , n in FIG. **9** contain the actual payload being transferred. Therefore, most of the capacity of the frame is allocated to the payload. The numbered blocks correspond, depending on the configuration, to e.g. an 8 kbit/s or a 16 kbit/s channel.

The SMF frame of FIG. **9** comprises two frame flags, one at each end of the frame. The frame flags are needed to indicate the start and finish of a variable-length SMF frame. The frame length depends on the number of subframes transferred in it. Using the flag, the statistical multiplexer of the receiving unit synchronizes to the frame and is able to read it as intended.

To interpret the payload carried in the SMF frame, information on the structure of the frame, i.e. the positioning of the channels in the SMF frame, is transferred in the allocation field of the frame. Based on this control information, the statistical demultiplexer is able to insert the bits transferred in the frame to the correct position in e.g. a PCM frame. Allocation messages are placed into the allocation field of the SMF frame, which are possibly numbered with a modulo8 function (not shown in the figure), whereby it is possible to refer to allocation messages already sent in e.g. a fault situation. The allocation messages consist of messages of e.g. the form: 'channel number, transmission block number, transmission block number', which means that the transfer of channel **1** in blocks **3** and **9** would be expressed in the allocation message in the form: '**ch1; 3, 9**'. From the allocation message, the channel number takes up one octet (8 bits), with which 256 different half-rate channels can be addressed. For addressing the transmission blocks of half-rate 8 kbit/s channels, a further octet is needed, and for

full-rate channels as much as two octets, depending on the coding method. If the SMF frame is configured to 16 kbit/s transmission blocks, only, mere 7 bits are required to address the block. As an allocation message is only included into the SMF frame that starts the forwarding of subframes, a considerable capacity saving is achieved when compared to a situation where the channel number is attached to every subframe forwarded. The subframe allocations of an SMF frame may be independent in different transmission directions. If, however, the allocations are the same in both transmission directions, the allocation field is not necessarily required but for one transmission direction.

When an active speech period comes to an end, the channel allocation to the transmission blocks of the SMF frame does not necessarily last, if these allocated transmission blocks are needed for transfer of another active channel. In such a case, it is necessary to reallocate transmission blocks for the channel that has lost an allocation, e.g. for frames carrying SID comfort noise parameters according to the SACCH multiframe structure. Consequently, a plurality of allocation instances has to be transferred in the allocation field for the same channel, but only in the case of a number of activity periods.

The inventive statistical multiplexer must obtain unambiguous information on every subchannel carried in the SMF frame to be able to demultiplex the frame. Based on the information in the allocation message, each channel is inserted into the correct timeslot of the fixed-length frame, e.g. a PCM frame, in the demultiplexer. On the basis of the channel number transferred in the allocation message it is possible to deduct whether the channel is a signaling or traffic channel, and on the basis of allocated blocks, the capacity of the channel. In the allocation message, it is also possible to indicate removal of synchronization and/or control bits, whereby the demultiplexer can generate them. From the point of view of the receiving unit, the transmission blocks of the SMF frame are allocated to a specific channel until the transmitting unit sends a new allocation message in which they are allocated to another channel.

The allocation field's length is predetermined, and the information bits therein are advantageously protected, by e.g. CRC (Cyclic Redundancy Check), to detect possible transmission errors. On a transmission line of sufficient quality, the CRC procedure is not necessarily needed at all. The CRC check sum calculated from the allocation field bits is placed at the end of the allocation field, as shown e.g. in FIG. **9**. The CRC procedure detecting a transmission error is advantageous, because in the allocation field, information is transmitted on the basis of which the statistical demultiplexer disassembles the SMF frame. Erroneous allocation information would result in erroneous channel disassembly from the SMF frame, and therefore confusion for the entire duration of the communication. To the return direction, information on possible CRC errors is transferred in the ACK field, located for example at the beginning of the allocation field, as shown by FIG. **9**. The ACK bits, too, can be calculated to the CRC checksum. When the ACK field indicates a detected transfer error, the transmitting unit can resend the allocation messages of the corrupted allocation field, if they are still needed. The information of the corrupted allocation field is advantageously transmitted in the first possible SMF frame. The contents of the erroneous allocation message are not noted at the receiving end, but the SMF frame is demultiplexed on the basis of previous allocations. FIG. **10** shows as a signalling chart an SMF frame disassembly error caused by a corrupted allocation message. As a finite length of time is spent when the allocation messages contained in the corrupted allocation message are retransmitted, it is in the worst case possible to connect two TRAU subframes/channel to wrong bit positions into a fixed-length frame. e.g. a PCM frame according to G.704. The erroneous connecting naturally only concerns new allocations. The allocation message in the SMF1 frame of FIG. **10** is detected as having been corrupted during the transfer, whereby DEMUX sends information on the detected error to MUX in the ACK field, in the form of NAK (Negative Acknowledgement), for example. DEMUX demultiplexes the SMF1 frame on the basis of previous allocations. Prior to receiving the error information, MUX may hasten to send a new frame SMF2, which DEMUX still demultiplexes on the basis of previous allocations. It is not until SMF3 frame that MUX retransmits the allocation messages corrupted in the SMF1 frame. which means that DEMUX is able to correctly demultiplex the SMF frames from the reception of SMF3 on. To erroneously demultiplex two successive subframes of a channel into e.g. a PCM frame only causes some kind of disturbance in the traffic being forwarded, but as such does not endanger traffic transfer as a whole.

The maximum length of the SMF frame is limited by the available transfer capacity and repeat rate T1 at which the frame is transmitted. For example, if the interval between transmitting frames is T1=20 ms, the maximum length for the frame will be 39680 bits on a 2 Mbit/s bus, if the PCM frame timeslot **0** is allocated to synchronization bits according to G.704, as shown in FIG. **4**. If T1 is e.g. 5 ms, the TRAU frame is transferred in four successive SMF frames

The advantage of the inventive SMF frame is the small portion of identification information transferred in the frame in relation to the payload transmitted in the frame, which results in good transfer efficiency.

The following will describe in more detail the insertion of the payload into the inventive SMF frame and the associated allocation procedure, with reference to the example of FIG. **11**. FIG. **11** illustrates transfer of 16 kbit/s information of four TRAU frames to a PCM frame and from there to the inventive SMF frame. Two bits from each TRAU1–TRAU4 frame are transferred to a PCM frame according to G.704, for example. Prior to transferring the bits from the PCM frame to the SMF1 frame, the inventive activity checking is carried out for each channel, as described above. In the example of FIG. **11**, TRAU3 and TRAU4 frames are e.g. idle TRAU speech frames, whereby transmission blocks are allocated from the SMF1 frame only to the bits of the active channels' TRAU1 and TRAU2 frames

In the inventive allocation procedure, a position from the SMF frame is allocated to the activated channel, in which the subframes belonging to the channel are transferred as such without identification information. The allocation information of this position is transferred over the transmission bus to the receiving unit in an allocation field of the SMF frame in which information from the activated channel is first transferred. In the allocation field, a plurality of allocations may be transferred in the allocation messages of the allocation message. Each active period on each channel is allocated once only, i.e. the allocation message is sent only once per period over the transmission bus. As only new allocations are transferred in the allocation message, it cannot be used to deduct the SMF frame's utilization degree. In the SMF1 frame allocation field of the example of FIG. **11**, allocation messages 'ch1; **3,9**' and 'ch2; **1,12**' are transmitted, which indicate to the receiving unit that transmission blocks **3** and **9** have been allocated for transfer of information of channel **1** and correspondingly transmission

blocks 1 and 12 for channel 2. The subframes of these two channels are inserted into the payload field of SMF1 frame, in said transmission blocks. Subframes of other channels are inserted into the other transmission blocks of the SMF1 frame's payload field according to allocations made. The formed SMF1 frame is transmitted over the transmission bus to toe receiving unit which reads the new allocations made to the payload field from the allocation messages of the allocation field, and demultiplexes the payload field, subframes to a fixed-length transmission frame, e.g. a PCM frame, on the basis of the these new and the previous allocations, reversely from what is shown in FIG. **11**. The receiving unit also generates idle TRAU speech frames for channels **3** and **4**, if required by transmitting the fixed-length frames forward. The rest of the subframes of the TRAU1 and TRAU2 frames are transmitted in the subsequent SMF frames, without an allocation message in the SMF frames' allocation field.

FIG. **12** shows, in block diagram form, an embodiment of the statistical multiplexer carrying out the inventive statistical multiplexing. This multiplexer comprise an interface four PCM traffic **401** a checking unit **403**, a frame handling unit **405**, centralized control logic CNTRL and bus connection **407** for the SMF frames transferred. The checking unit **403** further comprises a switching unit SW for e.g. 16 kbit/s channels and signal processors DSP1–DSPn assigned for the channels. The frame handling unit **405** comprises a centralized RAM unit and a frame assembling unit FAD (Frame Assembler-Disassembler). The switching unit SW connects the signals received e.g. in a G.704 frame from PCM interface **401** to the correct signal processor DSP, each channel having its own DSP. Signal processor DSP recognizes the state of its channel and the traffic transferred therein from e.g. the synchronization pattern of the TRAU frame and the signalling frame's flag. According to the invention, the DSP requests the control unit CNTRL to allocate transfer resources to the activated channel from the SMF frame. The control unit CNTRL searches for as many free blocks in the SMF frame as required by the transfer rate of the activated channel. Also, when traffic ceases on the channel, signal processor DSP informs the CNTRL of this in order for the CNTRL to be able to detect unused blocks as it is searching for space in the SMF frame. Control unit CNTRL stores all existing allocations in memory. The DSP transfers the data carried on the active channel to the RAM unit in blocks having the length T1, i.e. in subframes, which are stored in the RAM unit into an address that the control unit has allocated to the channel. If needed, data can also be buffered in the RAM unit for later transmission. The FAD unit combines these subframes that are in the RAM unit into the inventive SMF frame that is transmitted to e.g. a 2 Mbit/s bus **407**, advantageously concurrently with its assembly. As traffic on the channel begins, the control unit CNTRL attaches the channels identification information, i.e. information on new allocations of the frame structure, to the allocation message of the SMF frame carrying the fist subframe of the active channel, for the receiving unit In the subsequent frames, this identification information is no longer transmitted. Once the activated channel's position has been allocated from the SMF frame, the allocation is valid until the control unit allocates the same blocks for use by another channel. Allocations are not released separately. In a normal case, a new allocation is not made before traffic on the first channel ceases, but in a congested state the active channel may temporarily lose its position in the middle of an activity period.

The centralized control logic CNTRL also supervises the multiplexer's operation. In particular, it monitors the number of subframes in the RAM unit. Consequently, congestion management can be arranged by e.g. commanding the signal processors DSP to momentarily interrupt the transfer of TRAU frames. In such a case. TRAU frames are lost in an equal manner on several channels, without totally preventing traffic on any one channel.

As the multiplexer packets the transferred data, it is possible to further enhance the transfer by incorporating data frame compression into the multiplexer, e.g. in accordance with recommendation V.42bis.

FIG. **13** shows, in block diagram form, an embodiment of the statistical multiplexer carrying out the inventive statistical multiplexing. This multiplexer comprises a bus interface **501** for the received SMF frames, a frame handling unit **503**, a regular-frame assembling unit **65**, centralized control logic CNTRL, a CCUFTC emulator EMUL and an interface for PCM traffic **507**. The frame handling unit **503** further comprises a frame demultiplexing unit FAD (Frame Assembler-Disassembler) and a centralized RAM unit The regular frame assembling unit **505** comprises signal processor DSP1–DSPn and a switching unit SW. The control unit in the statistical demultiplexer carrying out the inventive functionality controls the disassembly of an SMF frame on the basis of allocation information transferred in the allocation messages. The SMF frame received from bus interface **501** is demultiplexed in the FAD unit based on the allocation information of control unit CNTRL, and each channel's subframe is transferred to the RAM unit to the correct address. The subframes of the channels are transferred from the RAM unit to signal processor DSP allocated to each channel. CCU/TC emulator EMUL generates the bit stream required by transcoder TC and channel coding unit CCU onto the active channels during speech pauses when no idle data has been transferred on the bus, according to the invention. In such a case, emulator EMUL generates e.g. idle TRAU speech frames, during which the transmitting unit does not transmit bit streams on the quiet channel. Emulator EMUL generates, if need be, also the extra management bits, such as synchronization and/or control bits, removed in the transmitting unit The bit streams are inserted through the switching unit SW into the correct timeslot of e.g. a PCM frame and transmitted forward via Interface **507**.

The inventive statistical multiplexer and demultiplexer can be integrated, in which case some of the units shows in FIGS. **12** and **13** may be common. The statistical multiplexer and demultiplexer may also be integrated in a network element.

As regards the traffic transferred in the network, the inventive statistical multiplexing is transparent. The inventive SMF frame may be transferred over e.g. ATM (Asynchronous Transfer Mode), because the frame is transparent to the network.

The statistical multiplexer and bus may be dimensioned according to the average data amount transferred on the channel. The transmission line between the statistical multiplexer and demultiplexer is thus dimensioned smaller than the maximum traffic load. The highest transfer capacity saving is achieved when the base station connections are of low capacity and there is a plurality of them. Further, statistical multiplexing provides the biggest advantage in a case where the data transferred on the channels consists of bursts, i.e. when ts momentary bandwidth requirement is considerably higher than the average data rate.

The figures and the related description are only intended to illustrate the idea of the invention. The inventive multiplexing may vary in details within the scope of the claims.

Although the invention is in the above described mainly in connection with the GSM system, the method is generally applicable for use in telecommunications networks. The invention is particularly well suited for speech transfer in telephone traffic, but it may also be employed for transfer of other type of information, such as data transfer.

What is claimed is:

1. A method for carrying out statistical multiplexing in a telecommunications network which comprises a transmitting unit for transferring channels containing information from at least two traffic sources, a receiving unit and a transmission link between the at least two traffic sources, the information to be transmitted in the telecommunication system being transferred over the transmission link in transmission frames,

the method comprising:

assembling a variable-length transmission frame which comprises an allocation field and an information field,

identifying an active channel of channels from a plurality of traffic sources,

allocating an information field transmission block from the variable-length transmission frame to the information of the active channel to be transferred,

inserting the information of the active channel to be transferred into the allocated transmission block,

adding channel allocation information into an allocation field of the variable-length transmission frame in which the information of the active channel is for a first time continuous transfer,

maintaining the channel allocation for as long as a channel in question is continuously active, and

transmitting said variable-length transmission frame from the transmitting unit to the receiving unit over the transmission link.

2. A method as claimed in claim 1, further comprising:

inserting, in the transmitting unit, the information of the active channel to be transferred from a structurally regular fixed-length frame into said variable-length transmission frame, into a data transfer block allocated for use by the channel in question, and

transferring, in the receiving unit, the information of the active channel from the transmission block of the variable-length transmission frame to the structurally regular fixed-length frame.

3. A method as claimed in claim 1 in a telecommunications network which employs a discontinuous transmission feature in at least part of the traffic sources, further comprising:

distinguishing payload information and unnecessary bit stream on at least one channel of a traffic source,

inserting the payload only into the information field of the variable-length transmission frame.

4. A method as claimed in claim 3, wherein the unnecessary bit stream is generated among the payload in the receiving unit.

5. A method as claimed in claim 1, further comprising:

identifying traffic activity on channels of a plurality of traffic sources, and

inserting information of active channels into the information field of the variable-length transmission frame.

6. A method as claimed in claim 1, further comprising the steps of:

removing spare management bits from the information of the active channel to be transferred,

inserting the remaining bits into the transmission block allocated to the active channel, and

generating the removed spare management bits in the receiving unit.

7. A method as claimed in claim 6, wherein at least one synchronization bit is removed.

8. A method as claimed in claim 6, wherein at least one control bit is removed.

9. A method as claimed in claim 1, further comprising:

assembling a variable-length transmission frame prior to transmitting the variable-length transmission frame to the transmission link.

10. A method as claimed in claim 1, further comprising:

assembling a variable-length transmission frame having a fixed length allocation field.

11. A method as claimed in claim 10, further comprising:

protecting the allocation field to detect transmission errors, and

retransmitting information of an erroneous allocation field in the allocation field of a new transmission frame.

12. A statistical multiplexer to multiplex information of a plurality of channels in a telecommunications network, the multiplexer comprising an interface for receiving traffic transferred in a fixed length frame, wherein the statistical multiplexer further comprises:

a checking unit for identifying an active channel of channels from a plurality of traffic sources,

a frame handling unit for assembling a variable-length transmission frame comprising an allocation field and an information field,

centralized control logic for controlling a functionality of the checking unit and the frame handling unit, for handling the allocation field of the variable-length transmission frame, for storing existing allocations in memory, and for maintaining channel allocation for as long as a channel in question is continuously active, and

an interface for transferring the variable-length transmission frames.

13. A statistical multiplexer as claimed in claim 12, wherein the checking unit comprises:

a switching unit for switching fixed length frames of single channels, and

signal processors for identifying active periods of at least one channel.

14. A statistical multiplexer as claimed in claim 12, wherein the frame handling unit further comprises:

a memory unit for buffering information of active channels, and

a frame assembling unit for inserting information into a variable-length transmission frame.

15. A statistical multiplexer as claimed in claim 12, wherein the centralized control logic is adapted to allocate transmission blocks of the variable-length transmission frame and to incorporate an allocation into said variable-length transmission frame.

16. A statistical demultiplexer for disassembling multiplexed information of a plurality of channels in a telecommunications network, the demultiplexer comprising an interface for transferring fixed-length frames, the statistical demultiplexer further comprises:

an interface for receiving variable-length transmission frames,

a frame handling unit for disassembling the variable-length transmission frames comprising an allocation field and an information field,

a regular frame assembling unit for assembling fixed-length frames, and

centralized control logic for controlling a functionality of the frame handling unit and the regular frame assembling unit, and for maintaining channel allocation for as long as a channel in question is continuously active.

17. A statistical demultiplexer as claimed in claim **16**, wherein the demultiplexer further comprises an emulator for generating bits among information received in the variable-length transmission frame.

18. A statistical demultiplexer as claimed in claim **16**, wherein the frame handling unit comprises:

frame disassembling unit for disassembling information from the variable-length transmission frame, and

a memory unit for buffering the demultiplexed information.

19. A statistical demultiplexer as claimed in claim **16**, wherein the regular frame assembling unit comprises:

signal processors for processing the bit stream of each channel, and

a switching unit for inserting single channels into the fixed-length frames.

20. A statistical demultiplexer as claimed in claim **16**, wherein the centralized control logic is adapted to manage allocations of transmission blocks of the variable-length transmission frame, and to control an information disassembly from said variable-length transmission frame according to the allocations.

* * * * *

US006658512B1

(12) **United States Patent**
Gokulrangan

(10) **Patent No.: US 6,658,512 B1**
(45) **Date of Patent: Dec. 2, 2003**

(54) **ADMISSION CONTROL METHOD FOR DATA COMMUNICATIONS OVER PERIPHERAL BUSES**

(75) Inventor: **Venkat R. Gokulrangan**, Hillsboro, OR (US)

(73) Assignee: **Intel Corporation**, Santa Clara, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 502 days.

(21) Appl. No.: **09/671,236**

(22) Filed: **Sep. 28, 2000**

(51) **Int. Cl.**[7] **G06F 13/362**; H04J 3/14

(52) **U.S. Cl.** **710/117**; 710/107; 709/227; 370/232

(58) **Field of Search** 709/227–229; 370/230–234; 710/107–125

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,289,462	A	*	2/1994	Ahmadi et al.	370/232
5,347,511	A	*	9/1994	Gun	370/255
5,533,009	A	*	7/1996	Chen	370/232
5,533,205	A	*	7/1996	Blackledge et al.	710/117
5,568,165	A	*	10/1996	Kimura	345/547
5,594,717	A	*	1/1997	Watanabe et al.	370/232
5,805,599	A	*	9/1998	Mishra et al.	370/468
5,815,492	A	*	9/1998	Berthaud et al.	370/234
5,832,300	A	*	11/1998	Lowthert	710/33
5,896,413	A	*	4/1999	Yoshimura et al.	375/219
5,940,370	A	*	8/1999	Curtis et al.	370/231
5,982,748	A	*	11/1999	Yin et al.	370/232
5,982,750	A	*	11/1999	Tabe et al.	370/233
5,986,714	A	*	11/1999	Marino	375/240.05
6,084,955	A	*	7/2000	Key et al.	379/220.01
6,145,040	A	*	11/2000	LaBerge et al.	710/107
6,215,768	B1	*	4/2001	Kim	370/230
6,292,466	B1	*	9/2001	Droz	370/232
6,295,516	B1	*	9/2001	Takeyasu	703/13
6,359,862	B1	*	3/2002	Jeffries et al.	370/232
6,359,863	B1	*	3/2002	Varma et al.	370/232
6,359,889	B1	*	3/2002	Tazaki et al.	370/395.61
6,400,685	B1	*	6/2002	Park	370/232
6,438,141	B1	*	8/2002	Hanko et al.	370/477
6,459,681	B1	*	10/2002	Oliva	370/232
6,487,170	B1	*	11/2002	Chen et al.	370/231
6,490,249	B1	*	12/2002	Aboul-Magd et al.	370/232
6,542,467	B2	*	4/2003	Umayabashi	370/236

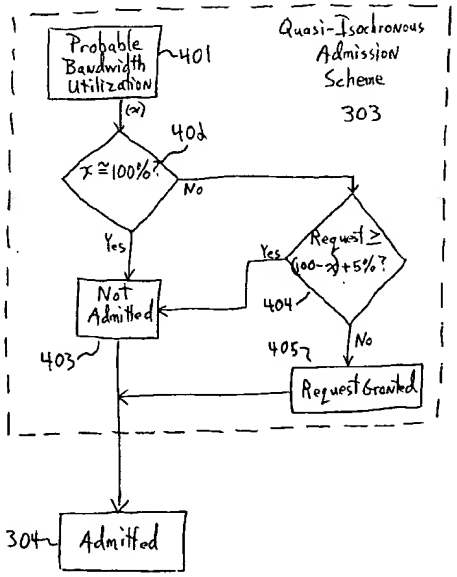
* cited by examiner

Primary Examiner—Sumati Lefkowitz
(74) *Attorney, Agent, or Firm*—Antonelli, Terry, Stout & Kraus, LLP

(57) **ABSTRACT**

A method adaptively controls admission of a plurality of resources to a peripheral bus having a finite bandwidth. It monitors the actual data transfer rate of each resource having a bandwidth guarantee on the peripheral bus for data communications. Upon receiving a request for admission to the peripheral bus from an additional resource, a utilization value representative of the extent to which the plurality of resources utilizes the guaranteed bandwidth on the peripheral bus is obtained. The additional resource is admitted to the peripheral bus if the utilization value indicates that the amount of unutilized bandwidth on the peripheral bus is approximately sufficient to satisfy the data transfer rate of the additional resource.

29 Claims, 4 Drawing Sheets



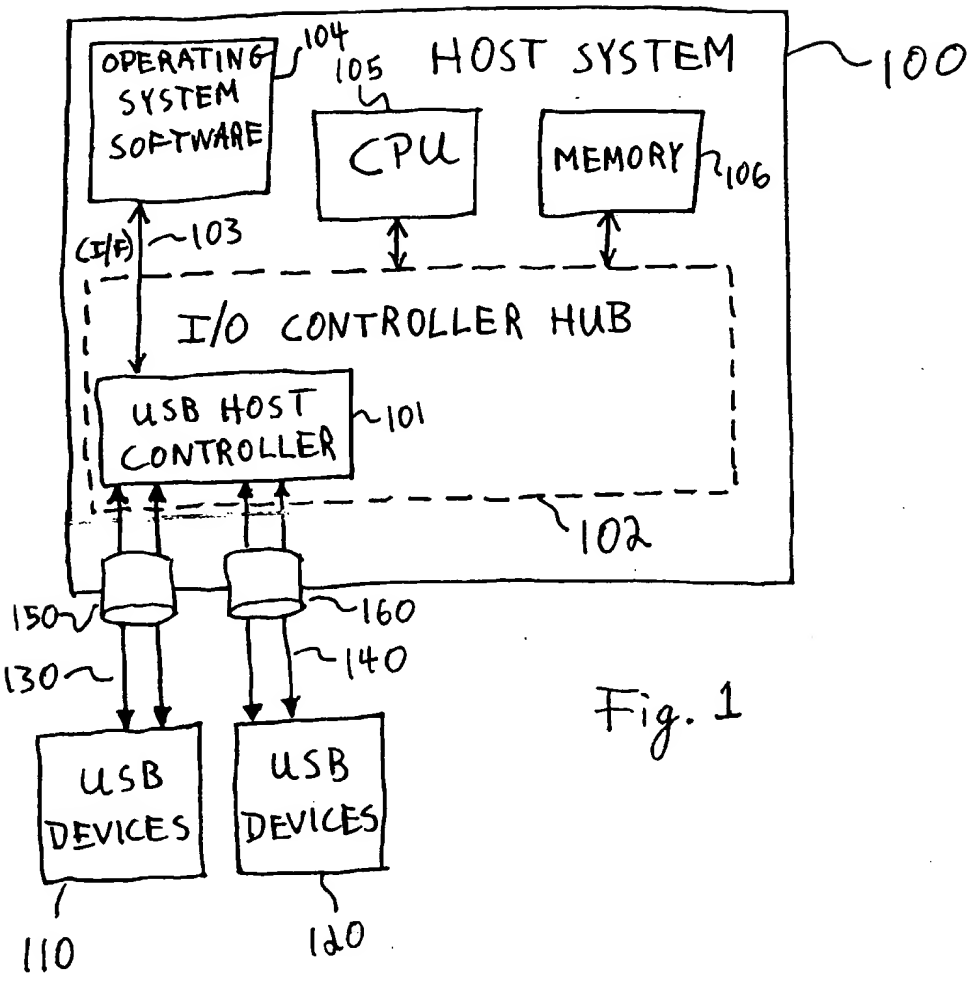

OPERATING SYSTEM SOFTWARE
104
105
HOST SYSTEM
100
CPU
MEMORY
106
C(F)
103
I/O CONTROLLER HUB
USB HOST CONTROLLER
101
102
150
160
130
140
USB DEVICES
USB DEVICES
110
120

Fig. 1

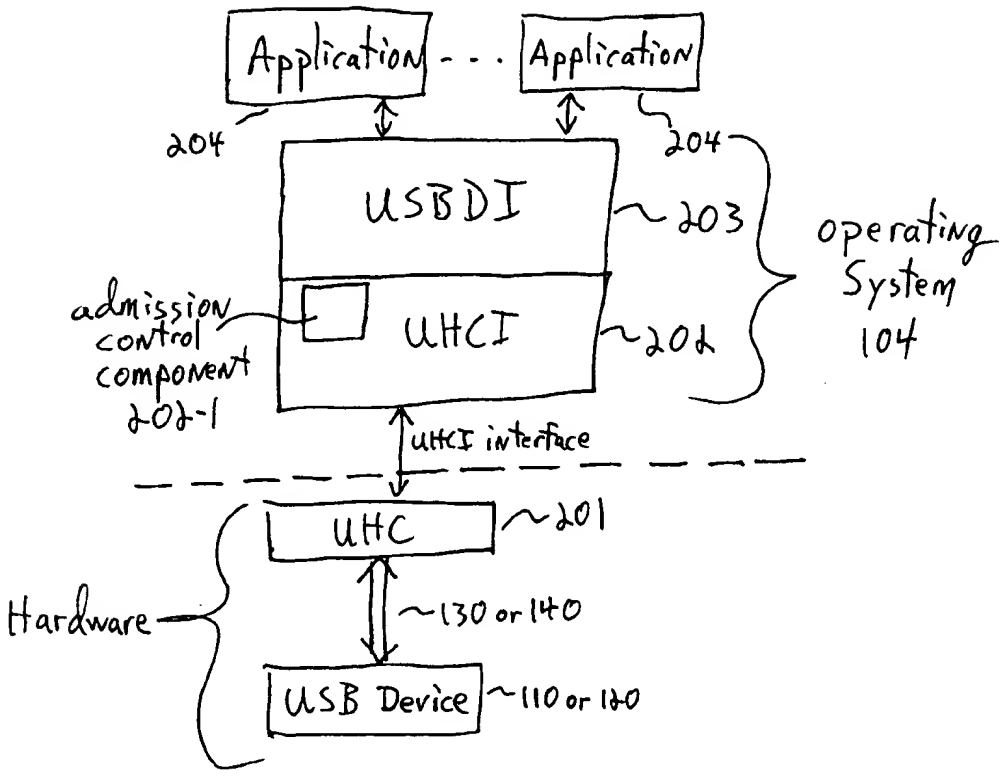

Application
Application
204
204
USBDI
203
operating System 104
admission control component 202-1
UHCI
202
UHCI interface
UHC
201
Hardware
130 or 140
USB Device
110 or 120

Fig. 2

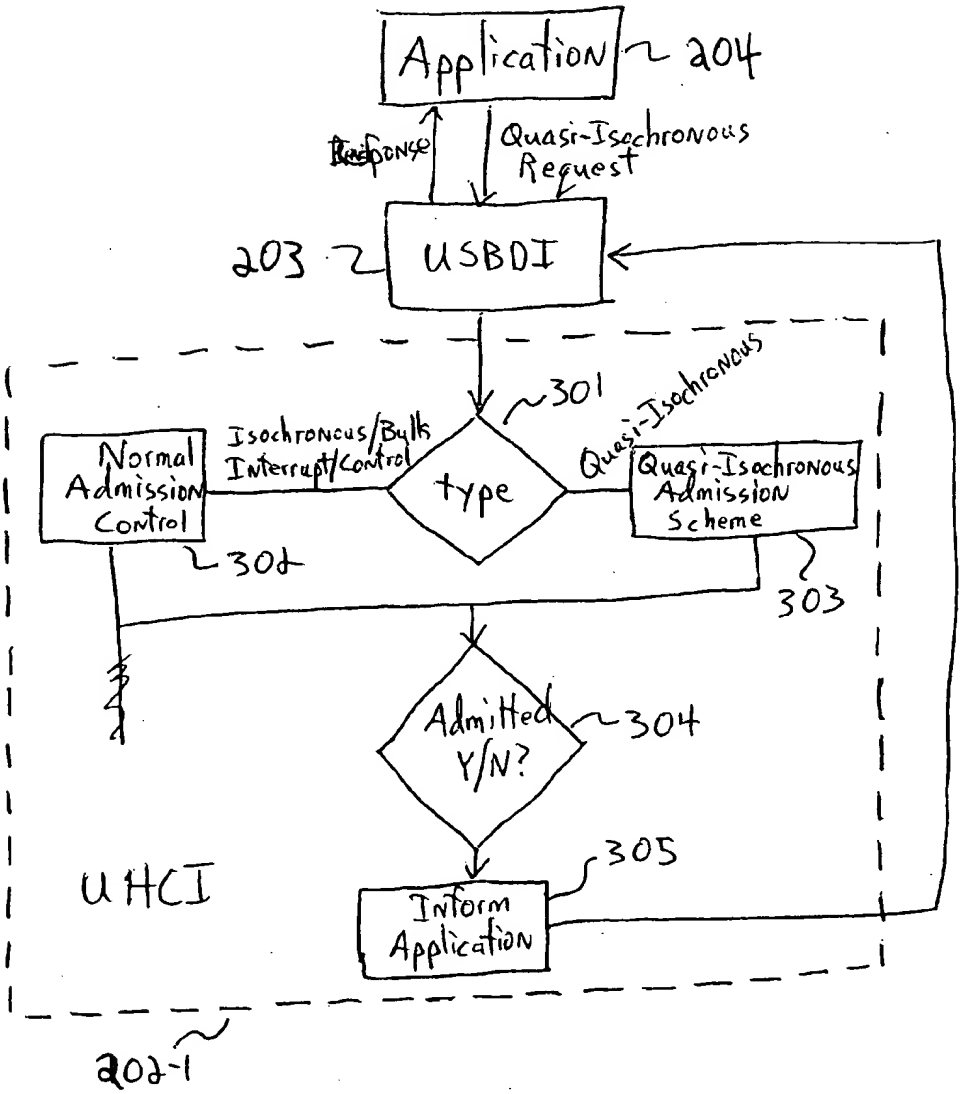

Application
204
Response
Quasi-Isochronous Request
203
USBDI
301
Isochronous/Bulk Interrupt/Control
Quasi-Isochronous
type
Normal Admission Control
302
Quasi-Isochronous Admission Scheme
303
Admitted Y/N?
304
305
Inform Application
UHCI
202-1

Fig. 3

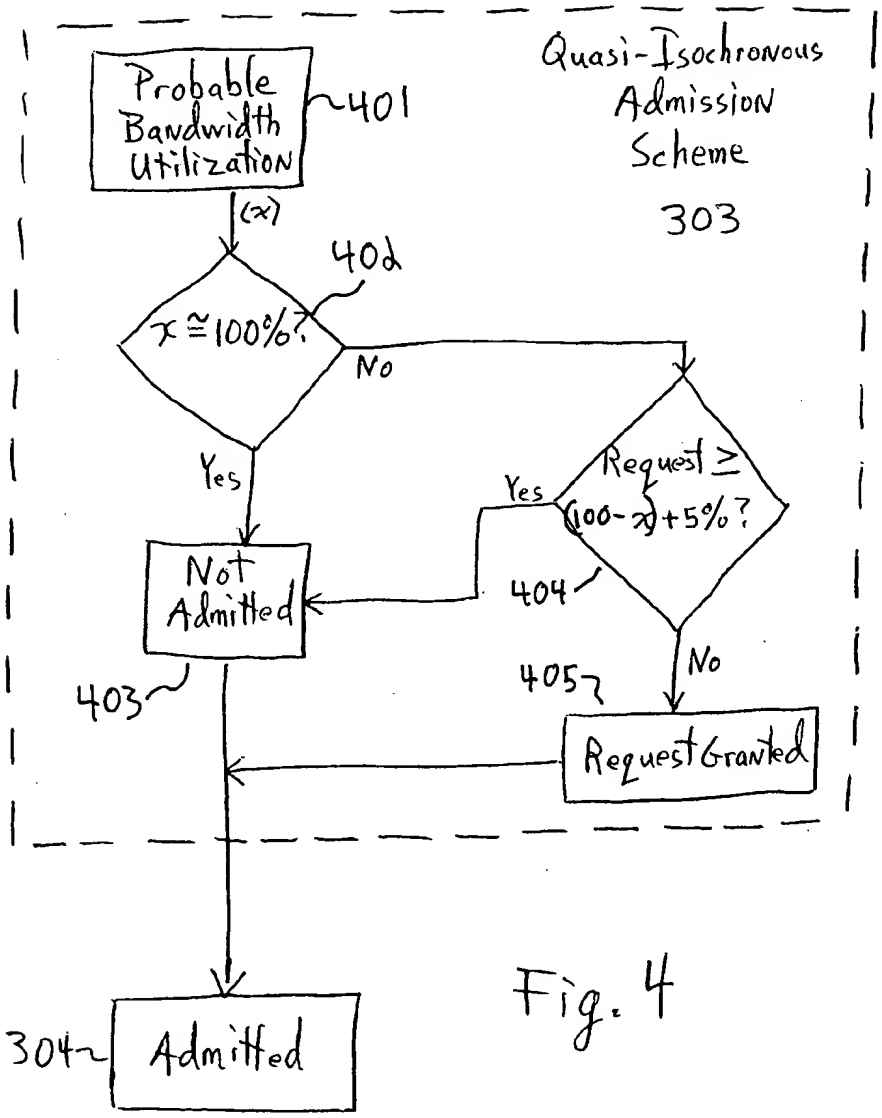

Quasi-Isochronous Admission Scheme
303
Probable Bandwidth Utilization
401
(x)
402
x ≅ 100%?
No
Yes
Request ≥ (100-x)+5%?
Yes
404
405
No
Not Admitted
403
Request Granted
304
Admitted

Fig. 4

ADMISSION CONTROL METHOD FOR DATA COMMUNICATIONS OVER PERIPHERAL BUSES

BACKGROUND

1. Field of the Invention

This invention relates generally to data communications. In particular, the present invention relates to methods of controlling the admission of a plurality of data communications on a peripheral bus for a computing device having a finite bandwidth.

2. Description of the Related Art

Communication links, such as peripheral buses in a network, can transfer data for a plurality of different computing devices, software applications and other resources resident on the network. The data from a computing device, application or other resource can be transferred over a peripheral bus using a Class of Service (CoS) or other method which provides timeline, maximum error rate or other Quality of Service (QoS) guarantees. Some transfer classes reserve and pre-allocate part of the bandwidth of the peripheral bus in advance for continuous and constant guaranteed availability to the resource. (Although the peripheral bus may be any kind of a cable, fiber optic, wired connection or wireless connection, the term "bandwidth" is nevertheless used herein to refer to the amount of data which can be transferred per unit of time, typically in bits per second (b/s) or bytes per second (B/s), and does not refer to the frequency range over which the peripheral bus transfers data.)

However, resources requesting guaranteed bandwidth do not always use all of the bandwidth they request for various reasons. For example, the amount of bandwidth requested is usually that amount which is sufficient to handle the worst case scenario (typically the expected peak data transfer rate). For example, a multimedia application receiving data through a digital subscriber line (DSL) controller might request 50% of a approximately 1.5 mbps throughput rate as a worst case scenario, where the best scenario scenario might be 25% of the throughput rate. Devices, application and resources do not always receive data at the maximum possible rate and therefore the average data transfer rate is often less than the peak data transfer rate. Also, the nature of some resources may require varying degrees of encoding/compression of data and thus cause the actual data transfer rate to vary substantially from the average data transfer rate. This is especially true of devices, such as video cameras, and applications, such as multimedia content player, which utilize real-time adaptive compression or decompression algorithms. This is a problem for communication systems which do not allow variable bit rate transfers. The reserved bandwidth pre-allocated to the device, application or resource will frequently go unused and wasted. The proportion of wasted bandwidth may very well increase significantly if the amount of bandwidth reserved by one or more devices, applications or resources increases and/or if the amount of bandwidth on the peripheral bus increases.

Furthermore, in some computer systems and networks, the devices, applications and resources are merely expected to be "good citizens" which will request only as much bandwidth as they will need and allow as much bandwidth as possible to remain to be used by other devices, applications or resources. See, for example, "An Analysis of Throughput Characteristics of Universal Serial Bus" by John Garney, Intel Corporation, Dec. 6, 1996. Such a computer system or network may not have any protections against faulty or rogue devices, applications or resources which inappropriately reserve bandwidth or reserve an inordinately large fraction of bandwidth. Therefore, the bandwidth on a peripheral bus may be hoarded by a resource or otherwise under-utilized.

BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding and appreciation of the foregoing and of the attendant advantages of the present invention will become apparent from the following detailed description of example embodiments of the invention in conjunction with the accompanying drawings. While the foregoing and following written and illustrated disclosure focuses on disclosing example embodiments of the invention, it should be clearly understood that the same is by way of illustration and example only and the invention is not limited thereto.

The following represents brief descriptions of the drawings, wherein:

FIG. **1** is a generalized block diagram of an exemplary computer system in which an example embodiment of the invention may be implemented.

FIG. **2** illustrates an example of the hardware and software layers related to the example embodiment of the invention in the exemplary computer system of FIG. **1**.

FIG. **3** illustrates an exemplary admission control method carried out in the Universal Host Controller Interface (UHCI) layer of FIG. **2** according to an example embodiment of the invention.

FIG. **4** illustrates an exemplary quasi-isochronous admission scheme carried out in the exemplary admission control method shown in FIG. **3**.

DETAILED DESCRIPTION

This application describes various example embodiments of the invention. Generally, the example embodiments involve a modified Universal Serial Bus (USB) communications system connecting a host computer system with a collection of modified USB devices. Typically, corresponding pieces of software and hardware in the host computer system and a USB device enable them to perform a function. However, the invention is not limited in its implementation to the USB example embodiments described in this application or to any other kind of communications system. Examples of other possible peripheral buses include IEEE 1394 and Firewire. Examples of possible communication systems include a wide area network (WAN), a campus area network (CAN), a metropolitan area network (MAN), a global area network (GAN), a storage area network (SAN), and a LAN network, including versions of Gigabit Ethernet, FDDI (Fiber Distributed Data Interface), Token Ring, Asynchronous Transfer Mode (ATM), Fibre Channel and Wireless. Of course, a wide variety of implementations, arrangements and configurations of all types of data communication systems and networks may be used with the invention.

This application also describes various example methods of adaptive admission control of data communications over peripheral buses. These example methods may be used in the example modified USB communication systems in this application or in any other suitable communications system. In any case, the scope of the invention should be determined by reference only to the claims contained in any patent issuing from this application and is not limited by any of the examples described herein.

For the sake of simplicity, discussions will concentrate mainly on an example modified USB data communication

system, the general architecture of which is shown in the block diagram of FIG. 1, even though the scope of the present invention is not limited thereto. While a host computer system **100** with operating system software, at least one central processing unit (CPU) **105** and memory **106** is utilized in the example embodiment shown in FIG. **1**, the present invention is applicable for use with all types of processing devices and systems, including any device or system which may be linked together with other disparate processing devices or systems such as computers, servers, peripherals, and storage devices through a peripheral bus in a communications network. The host system may consist of a processing system with limited resources, so long as it is able to support a USB host controller. The USB host controller may be located in an Input/Output Controller Hub (ICH) **102** as shown in FIG. **1** (other elements located in ICH **102** are not shown for the sake of simplicity); on a motherboard along with CPU **105** and/or memory **106**; or on another board, such as a printed circuit board. The location and packaging of the USB host controller **101** is not limited to any of the mentioned examples or limited in any other way. Conversely, the host system may be a large computer system containing a plurality of processors, storage units, etc, such as a server complex. Such a large computer system may have a plurality of USB host controllers, either combined or on separate units. Similarly, any type of USB device may be connected in the example embodiments described herein. Examples of possible USB peripheral devices include, but are not limited to, disk drives, printers, modems, keyboards, mice, pointing devices, video cameras, and audio devices.

In the example embodiment of FIG. **1**, two separate groups of USB devices **110** and **120** are interconnected to the USB host controller **101** of a host computer system **100** through respective USB buses **130** and **140**. The USB host controller **101** controls communications on the USB buses **130** and **140**, and transfers data between the USB buses **130** and **140** and the remainder of the host computer system **100**. USB host controller **101** has a root hub (not shown) in it or directly attached to it. The root hub has a port **150** (called a root port) associated with USB bus **130** which is the physical connection between USB bus **130** and USB host controller **111** and a port **160** associated with USB bus **140** which is the physical connection between USB bus **140** and USB host controller **111**. USB bus **130** connected to root port **150** (and USB bus **140** connected to port **160** of the root hub) is formed by the combination of a cable along with a voltage and ground line (not shown). Although only two ports are shown in FIG. **1**, the root hub can have up to six ports and six corresponding USB buses rather than the two ports **150** and **160** and two USB buses **130** and **140** shown in FIG. **1**.

With the addition of additional equipment (not shown), such as one or more "downstream" (non-root) hubs attached to root port **130** and/or port **140** of the root hub and/or other non-root hubs, USB host controller **101** may support up to **127** connected devices or functions. There is one "upstream" port and up to six "downstream" ports on non-root hubs, with each one of the downstream ports connecting the non-root hub to another hub or to a USB device through a USB cable. A USB hub may be a stand-alone device or it may be included within the chassis of a USB device, such as a keyboard. With operating system software **104** such as Windows 98® or Windows 2000® from Microsoft Corporation of Redmond, Wash. in host computer system **100**, it is possible to plug any USB device into any USB port on any connected USB hub and the operating system software **104** will, upon activation of the host computer system **100**, identify the USB device and interface to it. The user does not need to be concerned with setting up various communications parameters when connecting a USB device.

When USB host controller **101** receives data from any connected USB device, such as one of the USB devices in groups **110** and **120**, it sends a message to the operating system software **104** of the host system **100** over an interface **103**. Further details of the example embodiment related to the interaction between USB host controller **101** and the software in host processing system **100** are illustrated in FIG. **2**. When appropriate, like reference numerals and characters may be used in the various figures to designate identical, corresponding or similar components in differing figures. Further, in the detailed description of the various figures, exemplary sizes/mode/s/values/ranges may be given, although the present invention is not limited to the same. Also, well-known power and other connections to the components are not be shown for simplicity of illustration and discussion, and to avoid obscuring the invention.

In the example of FIG. **2**, USB host controller **101** is a Universal Host Controller (UHC) **201** and the interface **103** between UHC **201** and the software of host processing system **100** is the Universal Host Controller Interface (UHCI), Revision 1.1, March 1996. UHC **201** may be a stand-alone integrated circuit (IC) or it may be included as part of an I/O controller hub (ICH) having other I/O interfaces or it may be implemented in some other manner. A portion of UHC **201** is software or firmware provided in a chipset, an ICH, random access memory (RAM), read-only memory (ROM), programmable read-only memory (PROM), erasable programmable read-only memory (EPROM), or any other suitable form of storage in the host system. (Alternatively, USB host controller **101** may be an Open Host Controller (OHC) and the interface between the OHC and the software of host processing system **100** may be an Open Host Controller Interface (OHCI). In UHCI, most of the functions are carried out in software or firmware so that the hardware has lower gate counts and is cheaper to manufacture. On the other hand, in OHCI, more of the functions are carried out in the hardware, which means that OHCI is relatively more expensive.)

UHC **201** transfers data back and forth between a USB buses **130** and **140** and a system bus of host computer system **100**. According to the example embodiments of the invention, the operating system software **104** of host processing system **100** preferably includes a UHCI software layer **202** which is a modification of the UHCI layer described in the specifications for the Universal Host Controller Interface (UHCI), Revision 1.1, March 1996. Although UHCI layer **202** is a modified software layer, it nevertheless communicates with UHC **201** over the system bus of host computer system **100** in a manner in accordance with the UHCI specifications. (UHCI layer **202** need not be implemented in operating system software **104**. It could, for example, be implemented in an ICH or other I/O controller or it could be stored in an EEPROM implemented on a circuit board or I/O card. UHCI layer **202** may be stored at the time of manufacture or assembly of host computer system **100** or it may be later downloaded into host computer system **100**.) A USB Device Interface (USBDI) layer **203** provides an application programming interface (API) to UHCI layer **202** and one or more software applications **204**.

Regardless of the implementation of UHC **201** and UHCI layer **202**, they support classes of data communications according to USB 1.1 and/or USB 2.0 in which bandwidth is guaranteed as well as classes of data communications in which the bandwidth is not guaranteed. Non-guaranteed

USB data transfers include interrupt transfers and bulk transfers. Guaranteed USB data transfers include control transfers and isochronous transfers. Control transfers are used for handshaking, initialization of USB devices, etc. Interrupt transfers are on-demand transfers of small amounts of data. Since control transfers and interrupt transfers involve relatively small amounts of data, they do not typically present any bandwidth utilization problems. In bulk transfers, the data to be sent is posted in a buffer and is eventually sent with the lowest priority, i.e., when other data packets are needed to meet the deadlines of other data transfers on the same stream. These transfers are typically used for devices, such as printers, which are not time-sensitive. Although bulk transfers may contain large amounts of data, they usually are not a problem since they are categorized as the lowest priority transfer.

On the other hand, an isochronous transfer is a class of data communications in which a percentage of the bandwidth of the USB bus, sufficient to complete requested data transfers from the device, application or resource, is reserved and guaranteed in advance of the data transfer in accordance with the output bit rate of the device, application or resource. The necessary bandwidth is specified by the requesting device, application or resource itself in a "request for admission" made to the host controller of the USB bus. It is important that adequate bandwidth is requested by the USB device, application or resource since any data provided at a bit rate faster than the specified bit rate will be dropped during transfer. For this reason, the USB device or application usually calculates the amount of requested bandwidth from its required timeline guarantees for data delivery.

When the "request for admission" is made, an admission controller for the corresponding USB bus deterministically checks to see if the available bandwidth on the corresponding USB bus is sufficient to satisfy the requirements of the requesting application. In the circumstance of multiple isochronous devices, the admission controller typically operates according to a First-Come-First-Serve (FCFS) policy in which requests are considered sequentially in the order received. Therefore, if one or more previous requests for admission have been granted, then the admission controller checks to see if the remaining bandwidth which has not been previously allocated and guaranteed can meet the requirements of the currently requesting resource. If it can, then the request for admission is granted. If it can't, then the request for admission is denied.

Regardless of the number of connected hubs and USB devices in the data communication system, USB host controller **101** can only transfer a finite amount of bandwidth which must be shared between all of the connected USB devices. For example, an optical disk drive (i.e., Compact Disc Read Only Media (CD-ROM), Compact Disc Read/Write Media (CD-RW), Digital Versatile Disc Read Only Media (DVD-ROM) or Digital Versatile Disc Random Access Media (DVD-RAM)), a modem or other communication device and/or various multimedia devices may be simultaneously connected to the same USB bus. If any one of these devices hoards the USB isochronous bandwidth, it may prevent a new device from being admitted even though it does not utilize all of the bandwidth it is allocated. At worst, a misoperating or rogue device (or application on a device) can "hijack" the entire USB bus without actually performing any transfer and not allow any other application to be admitted to transfer data over the USB bus.

In order to cope with the problem in which bandwidth is mis-allocated or is allocated in accordance with the peak transfer rate of true isochronous devices, the example embodiments of the invention use a new data transfer class which will be called "Quasi-isochronous" or "Q-chronous". This quasi-isochronous class provides only a "virtual" isochronous transfer guarantee as opposed to real isochronous transfer guarantee provided in the isochronous transfer class. Data is transferred only when the peripheral bus is not fully utilized by true isochronous transfers or when small data packets can be transferred in intervals left vacant by true isochronous transfers. On the other hand, during bursty transfers from true isochronous devices or during lack of availability of bandwidth, the data in a quasi-isochronous transfer is not transferred and is dropped, if necessary. This quasi-isochronous class has relaxed delivery constraints, which are bound to be best-effort in the presence of peak-rate true isochronous data transfers, but which lead to an efficient utilization of the isochronous bandwidth whenever possible.

To implement the quasi-isochronous transfer class to efficiently use the finite USB isochronous bandwidth, the example embodiments of the invention include an admission control component **202-1** utilizing a unique adaptive bandwidth allocation scheme, different than the conventional FCFS admission policy, for those USB devices which are capable of quasi-isochronous transfers when they request quasi-isochronous transfers. As shown in FIG. **3**, when **20** an admission request is received, such as from one of the applications **204** through USBDI layer **203**, the admission control component **202-1** determines whether the requested transfer is either: 1) one of a conventional isochronous, bulk, interrupt or control transfer request; or 2) a quasi-isochronous transfer request (step **301** in FIG. **3**). If the admission request is a conventional isochronous, bulk, interrupt or control transfer, then the conventional admission control method is performed (step **302**) to determine whether or not to admit the request. If the admission request is for a quasi-isochronous transfer, then a quasi-isochronous admission scheme is performed (step **303**). Both admission schemes provide a yes/no result (step **304**) and the admission control component informs the requesting application of the result (step **305**).

Preferably, UHCI layer **202** or admission control component **202-1** dynamically monitors the actual isochronous bandwidth being used on the corresponding USB bus. When a quasi-isochronous transfer request is received, admission control component **202-1** in UHCI layer **202** adaptively grants admission to the request using a fair-share adaptive scheduling algorithm in quasi-isochronous admission scheme **303** which is based at least in part on the "real available bandwidth" rather than only on the bandwidth presumed to be available because it has not been previously allocated to isochronous transfer requests. The quasi-isochronous admission scheme **303** in admission control component **202-1** utilizes the actual available bandwidth to allocate the remaining bandwidth to quasi-isochronous transfer requests in a manner which may or may not be loosely based on the FCFS policy. Although not shown in the drawings, alternative embodiments of the invention may utilize hardware (or software) in UHC **201** to dynamically monitor the actual available bandwidth and/or may include an admission control component performing the same adaptive admission policy for quasi-isochronous transfers as that of admission control component **202-1**.

While a plurality of different quasi-isochronous admission schemes may be used in accordance with this invention, FIG. **4** illustrates an example quasi-admission scheme **303** in which the actual available bandwidth determined by dynamically monitoring the USB bus is supplemented with

feedback control scheme(s), such as predicted transfer rate, to calculate a percentage value "(x)" representing the probable bandwidth utilization (step **401**) used in the the adaptive admission policy. The quasi-isochronous admission scheme first determines whether or not the probable bandwidth utilization is approximately equal to 100% (step **402**). If it is, then the quasi-isochronous transfer request is not admitted (step **403**). If the probable bandwidth utilization "x" is less than approximately 100%, then it is next determined (step **404**) whether or not the amount of bandwidth requested in the quasi-isochronous request is greater than the probable available bandwidth (100%−"x") plus a small margin (for example, 5%) as shown in FIG. **4**). If the amount of bandwidth in the quasi-isochronous transfer request is greater, then the request is not admitted (step **403**). If the amount of bandwidth in the quasi-isochronous transfer request is not greater, than the quasi-isochronous transfer request is granted (step **405**). The admitted yes/no result of quasi-isochronous admission scheme **303** is reported by step **304** in FIG. **3** in the same manner as the results for conventional admission control scheme **302**.

The advantages of the availability of the quasi-isochronous transfer class and quasi-isochronous admission scheme can be illustrated by an example. Assume that a first device or application has a peak transfer rate requirement equal to 70% of the bandwidth of the peripheral bus and that a second device or application has a peak transfer rate requirement equal to 30% of the bandwidth of the peripheral bus. Conventionally, a third device or application, requiring only 10% of the bandwidth, will not be admitted even if the first device or application normally transfers data taking up only 40% of the bandwdith and the second device or application normally transfers data taking up only 20% of bandwidth. In the example embodiments of the invention, the third device or application can be admitted to transfer data on the peripheral bus as a quasi-isochronous data transfer.

As would be appreciated by one of ordinary skill in the art, the calculation of probable bandwidth utilization (step **401**) of a USB bus can be carried out according to many different algorithms and with varying degrees of precision. For example, the calculation could engage in a detailed historical analysis of data transfers on the communication link with varying amounts of historical data being stored. The trade-off between the precision of the calculation of probable bandwidth utilization in step **401** and the demands on the resource(s) supporting the admission control component including quasi-isochronous admission scheme **303** should be taken into account. For example, if the admission control component is implemented in the UHC **201**, there may be limited amount of local memory available to store data. Also, if the admission control component is implemented in a UHCI layer **202** in operating system **104**, calculation of probable bandwidth utilization in step **401** could be slowed because of other tasks occurring in host computer system **101**.

Preferably, the raw data obtained by dynamically monitoring the peripheral bus is not stored in unaltered form. Instead the data is immediately processed to arrive at a representation or approximation of the bandwidth utilization on the peripheral bus. The monitored data could be processed with the result used to control transitions in a state diagram in which each state in the state diagram corresponds to a different level of bandwidth utilization (i.e., there could be 11 states corresponding to 0%, 10%, 20%, . . . 90%, 100% bandwidth utilization). The number of states and/or the frequency of monitoring and/or the length of time for which state information is stored could be determined in advance (for example, in a hardware implementation) or could be controlled and changed as a user preference (for example, through software controls when the admission control component is implemented in software). Once the state information is determined, the raw data may be discarded. The probable bandwidth utilization **401** at the time of a quasi-isochronous transfer request can be determined by examining the history of state transitions in the state diagram. While any design could be selected, sufficient resources should be available or dedicated to carry out quasi-isochronous admission scheme **303**.

The invention maximizes the bandwidth utilization of any peripheral bus link which does not support variable bit rate transfers, such as USB and IEEE 1394/Firewire, in the presence of multiple isochronous transfers or transfers of variable volumes of data from different devices or applications which have an advance guarantee of available bandwidth. By using the new quasi-isochronous transfer class described herein in the example of a USB bus, the invention allows the admission of devices and applications with lower timing constraints, and improves the data throughput of the data transfers, which would otherwise have to resort to bulk transfer, seriously compromising their delivery constraints. It is especially advantageous for communication devices such as modems, especially with emerging applications such as Voice over Internet Protocol (VoIP), and for multi-media applications, such as video, which using adaptive compression/decompression techniques but nevertheless are QoS sensitive.

Other features of the invention may be apparent to those skilled in the art from the detailed description of the example embodiments and claims when read in connection with the accompanying drawings. While the foregoing and following written and illustrated disclosure focuses on disclosing example embodiments of the invention, it should be understood that the same is by way of illustration and example only, is not to be taken by way of limitation and may be modified in learned practice of the invention. While the foregoing has described what are considered to be example embodiments of the invention, it is understood that various modifications may be made therein and that the invention may be implemented in various forms and embodiments, and that it may be applied in numerous applications, only some of which have been described herein. It is intended by the following claims to claim all such modifications and variations.

What is claimed is:

1. A communications system controlling the admission of resources to a peripheral bus in the communications system, the communications system comprising:

a peripheral bus having a finite bandwidth;

a plurality of resources, connected to the peripheral bus, and configured to send and/or receive data over the peripheral bus; and

a communications controller connected to the peripheral bus, said communications controller adapted to:

monitor the actual data transfer rate on the peripheral bus of each of said plurality of resources having a bandwidth guarantee on the peripheral bus for data communications;

obtain a utilization value representative of the extent to which the plurality of resources utilizes the guaranteed bandwidth on the peripheral bus upon receiving a quasi-isochronous request for admission to the peripheral bus from an additional resource; and

admit the additional resource to the peripheral bus if the utilization value indicates that the amount of unutilized bandwidth on the peripheral bus is approximately sufficient to satisfy the data transfer rate of the additional resource.

2. The communications system recited in claim 1, wherein if the additional resource is admitted but the unutilized bandwidth is insufficient to transfer all of the bits of the additional resource, then the extraneous bits are dropped without being transferred over the peripheral bus.

3. The communications system recited in claim 1, wherein the step of obtaining a utilization value comprises monitoring the data traffic on the peripheral bus with an admission control component on a host system.

4. The communications system recited in claim 3, wherein the admission control component manages a state diagram.

5. The communications system recited in claim 4, wherein the states of the state diagram respectively correspond to different degrees of bandwidth utilization.

6. The communications system recited in claim 5, wherein the admission control component maintains a historical analysis of the data traffic on the peripheral bus.

7. The communications system recited in claim 1, wherein the peripheral bus is a USB bus.

8. The communications system recited in claim 7, wherein one of the plurality of resources contains a codec and sends and/or receives data having a variable bit rate.

9. The communications system of claim 1, wherein said plurality of resources comprise a plurality of Universal Serial Bus (USB) devices and said peripheral bus comprises a USB bus.

10. The communications system of claim 1, wherein obtaining the utilization value is performed only for the quasi-isochronous request.

11. The communications system of claim 1, wherein monitoring the actual data transfer rate comprises monitoring actual isochronous bandwidth being used on the peripheral bus.

12. A method of adaptively controlling admission of a plurality of resources to a peripheral bus having a finite bandwidth, said method comprising:

for each one of a plurality of resources having a bandwidth guarantee for data communications on the peripheral bus, monitoring the actual data transfer rate on the peripheral bus of each one of said plurality of resources;

upon receiving a quasi-isochronous request for admission to the peripheral bus from an additional resource, obtaining a utilization value representative of the extent to which the plurality of resources utilizes the guaranteed bandwidth on the peripheral bus for data communications; and

admitting the additional resource to the peripheral bus if the utilization value indicates that the amount of unutilized bandwidth on the peripheral bus is approximately sufficient to satisfy the data transfer rate of the additional resource.

13. The method recited in claim 12, wherein if the additional resource is admitted but the unutilized bandwidth is insufficient to transfer all of the bits of the additional resource, then the extraneous bits are dropped without being transferred.

14. The method recited in claim 12, wherein the step of obtaining a utilization value comprises monitoring the data traffic on the peripheral bus with an admission control component on a host system.

15. The method recited in claim 14, wherein the admission control component manages a state diagram.

16. The method recited in claim 15, wherein the states of the state diagram respectively correspond to different degrees of bandwidth utilization.

17. The method recited in claim 16, wherein the admission control component maintains a historical analysis of the data traffic on the peripheral bus.

18. The method recited in claim 12, wherein said plurality of resources comprise a plurality of Universal Serial Bus (USB) devices and said peripheral bus comprises a USB bus.

19. The method recited in claim 12, wherein obtaining the value is performed only for the quasi-isochronous request.

20. The method recited in claim 12, wherein monitoring the actual data transfer rate comprises monitoring actual isochronous bandwidth being used on the peripheral bus.

21. A computer program stored in a tangible medium, the computer program, when executed, causing a processing system to carry out a method comprising:

for each one of a plurality of resources having a bandwidth guarantee for data communications on the peripheral bus, monitoring the actual data transfer rate on the peripheral bus of each one of said plurality of resources;

upon receiving a quasi-isochronous request for admission to the peripheral bus from an additional resource, obtaining a utilization value representative of the extent to which the plurality of resources utilizes the guaranteed bandwidth on the peripheral bus for data communications; and

admitting the additional resource to the peripheral bus if the utilization value indicates that the amount of unutilized bandwidth on the peripheral bus for data communications is approximately sufficient to satisfy the data transfer rate of the additional resource.

22. The computer program recited in claim 21, wherein if the additional resource is admitted but the unutilized bandwidth is insufficient to transfer all of the bits of the additional resource, then the extraneous bits are dropped without being transferred.

23. The computer program recited in claim 21, wherein the step of obtaining a utilization value comprises monitoring the data traffic on the peripheral bus with an admission control component on a host system.

24. The computer program recited in claim 23, wherein the admission control component manages a state diagram.

25. The computer program recited in claim 24, wherein the states of the state diagram respectively correspond to different degrees of bandwidth utilization.

26. The computer program recited in claim 25, wherein the admission control component maintains a historical analysis of the data traffic on the peripheral bus.

27. The computer program recited in claim 21, wherein the utilization value is performed only for the quasi-isochronous request.

28. The computer program recited in claim 27, wherein said specific type of transfer request comprises a quasi-isochronous transfer request.

29. The computer program recited in claim 21, wherein monitoring the actual data transfer rate comprises monitoring actual isochronous bandwidth being used on the peripheral bus.

* * * * *

US006611694B1

(12) **United States Patent**
Oltedal et al.

(10) **Patent No.: US 6,611,694 B1**
(45) **Date of Patent: Aug. 26, 2003**

(54) **ARRANGEMENT FOR IMPROVING THE SPEECH QUALITY, ESPECIALLY FOR VOIP (VOICE OVER IP) CALLS**

(75) Inventors: **Einar Oltedal**, Kolbjørnsvik (NO); **Harald Johansen**, Rykene (NO); **Johan Karoly Peter Galyas**, Täby (SE)

(73) Assignee: **Telefonaktiebolaget LM Ericsson (publ)**, Stockholm (SE)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/522,994**

(22) Filed: **Mar. 9, 2000**

(30) **Foreign Application Priority Data**

Mar. 10, 1999 (NO) 19991169

(51) **Int. Cl.**[7] **H04B 1/38**; H04M 1/00; H04L 12/28

(52) **U.S. Cl.** **455/560**; 455/561; 455/422; 370/351; 370/352

(58) **Field of Search** 455/560, 561, 455/422, 63, 67.1; 370/351, 356

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,061,566 A * 5/2000 Friman
6,295,302 B1 * 9/2001 Hellwig et al.
6,324,402 B1 * 11/2001 Waugh et al.

FOREIGN PATENT DOCUMENTS

WO	WO95/15665	6/1995
WO	WO96/16521	5/1996

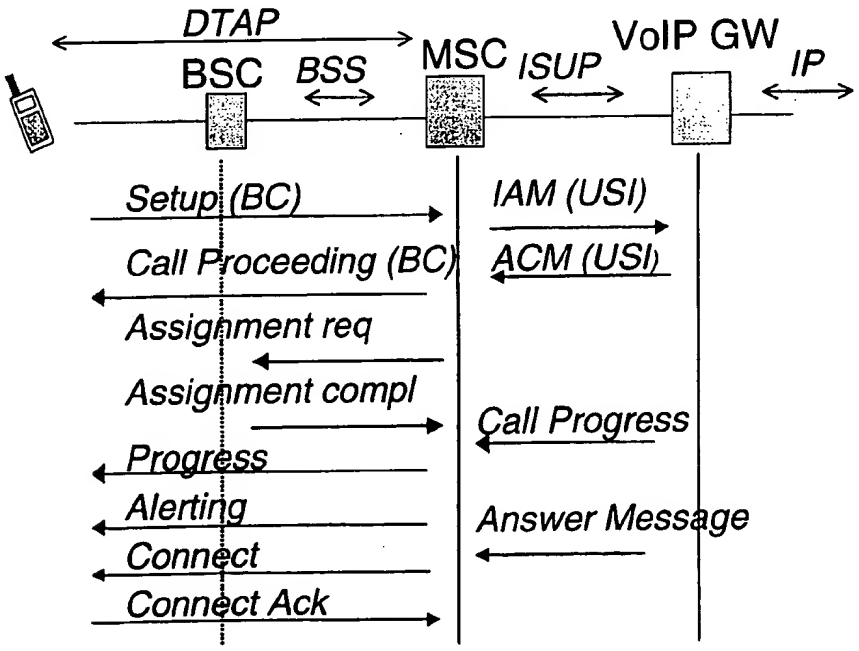
* cited by examiner

Primary Examiner—Thanh Cong Le
Assistant Examiner—Huy D Nguyen

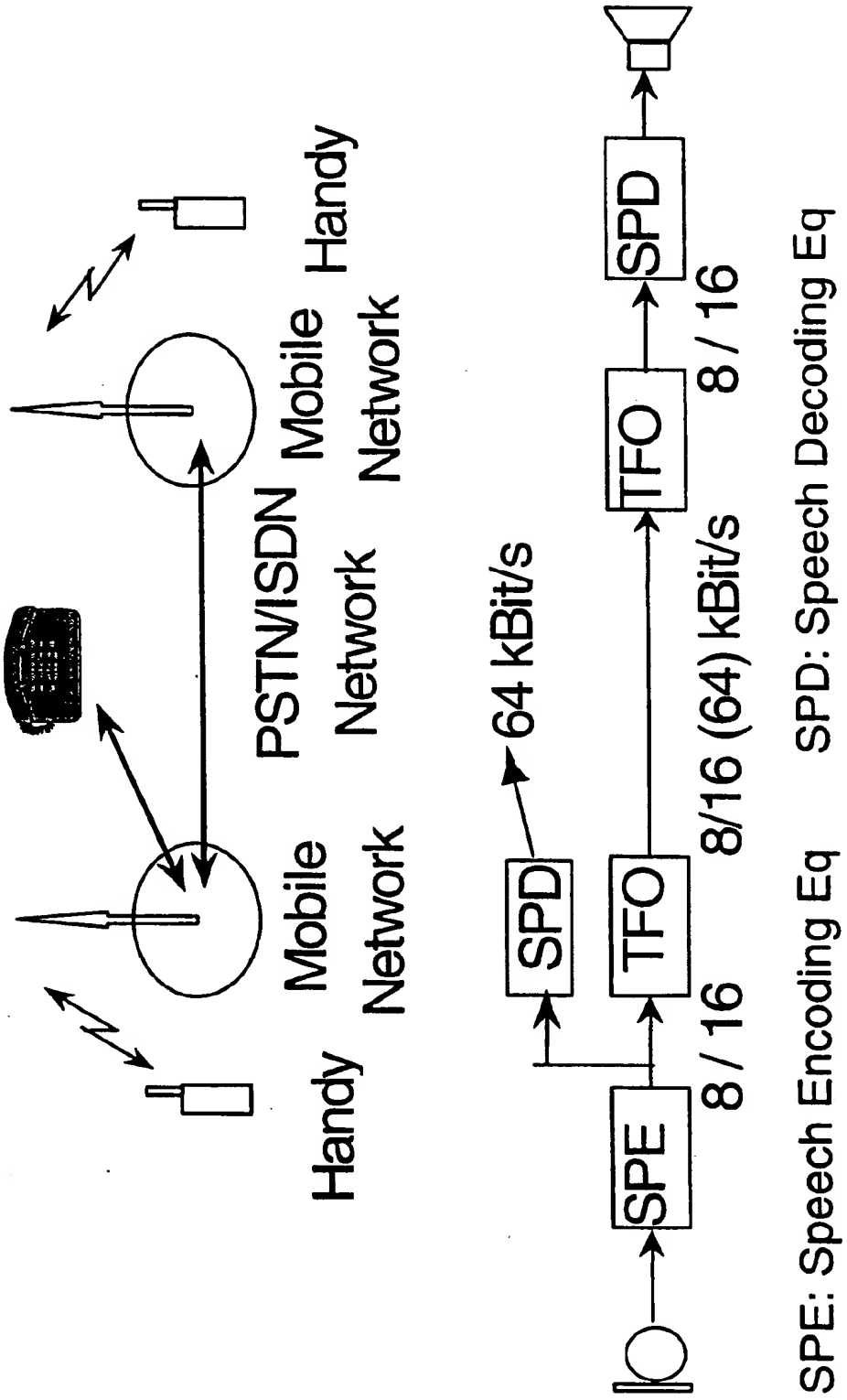
(57) **ABSTRACT**

The present invention relates to an arrangement for improving the speech quality, especially for VoIP (Voice over IP) calls, which arrangement comprises a Transceiver and Rate Adapter Unit (TRAU) in which an encoded speech signal from a Mobile Station (MS) is transcoded, and for the purpose of reducing the necessary encoding/decoding for thereby avoiding detorioration of speech quality, and also for the purpose of avoiding reduction in bandwidth, it is according to the present invention suggested that said arrangement comprises means for either putting the TRAU in a transparent mode or letting the TRAU be bypassed altogether.

3 Claims, 2 Drawing Sheets



Signalling sequences for TFO over IP.



SPE: Speech Encoding Eq SPD: Speech Decoding Eq

Figure 1 Principle of Tandem Free Operation.

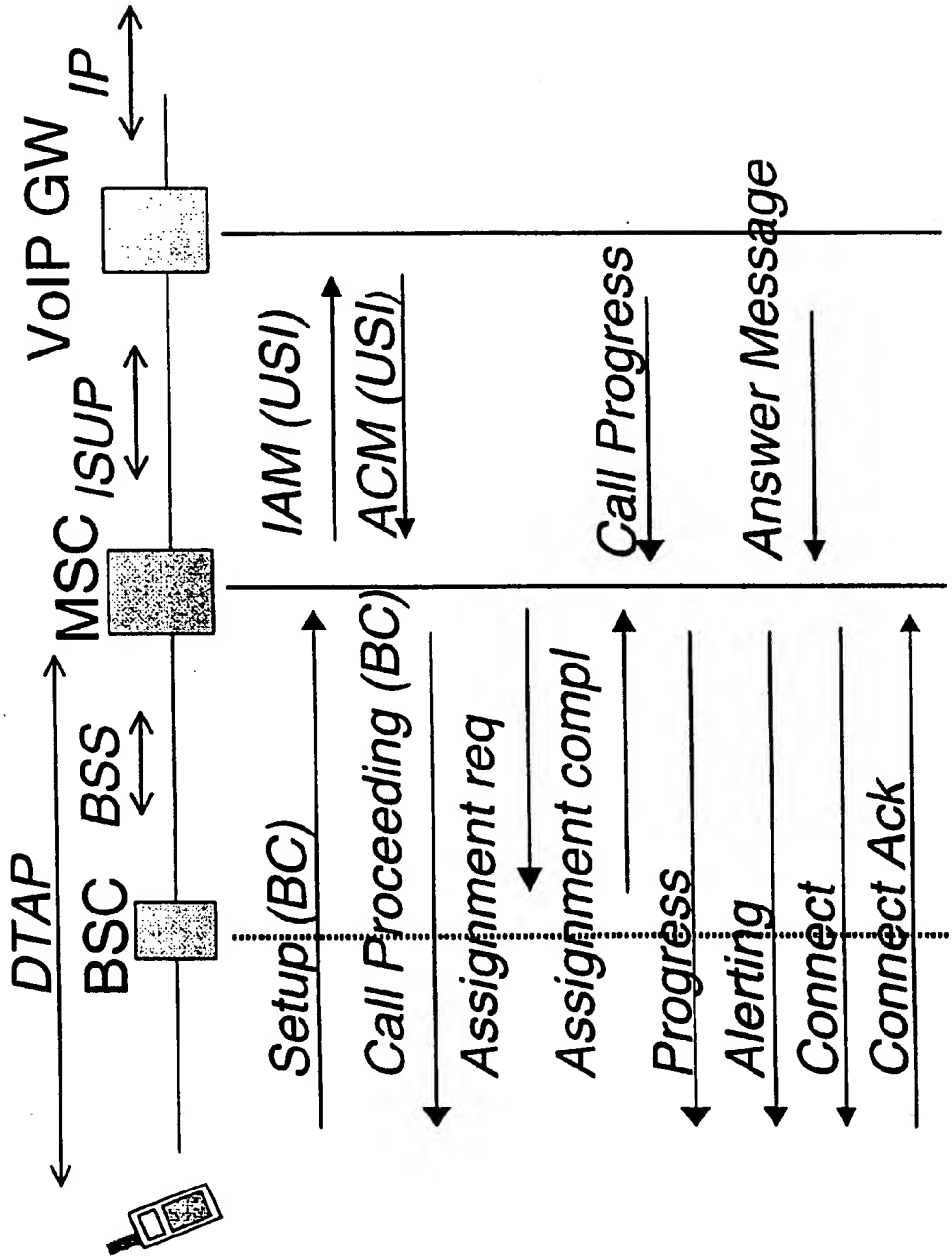


Figure 2 Signalling sequences for TFO over IP.

ARRANGEMENT FOR IMPROVING THE SPEECH QUALITY, ESPECIALLY FOR VOIP (VOICE OVER IP) CALLS

This application claims priority under 35 U.S.C. §§119 and/or 365 to 19991169 filed in Norway on Mar. 10, 1999; the entire content of which is hereby incorporated by reference.

FIELD OF THE INVENTION

The present invention relates to an arrangement for improving the speech quality, especially for VoIP (Voice over IP) calls, which arrangement comprises a Transceiver and Rate Adapter Unit (TRAU) in which an encoded speech signal from a Mobile Station (MS) is transcoded.

GENERAL BACKGROUND OF THE INVENTION

This is a proposal for how the speech quality can be enhanced for mobile VoIP calls. It is a well known problem that speech can be degraded for mobile calls when too many speech encodings/decodings are performed on the voice path.

There are mobile standards to overcome this for MS to MS calls, called Tandem Free Operation (TFO). The Japanese Personal Digital Cellular (PDC) system uses outband Mobile Application Part (MAP) signalling, called codec through. In Global System For Mobile Communication (GSM) there is an emerging standard, TS 04.53, while there is no TFO planned for Digital-Advanced Mobile Phone Service (D-AMPS).

These TFO standards were developed for MS to MS calls, and they do not address in an optimal way TFO over IP.

The solution described here has the following main benefits:

- Improved speech quality for mobile VoIP calls compared to ordinary mobile calls
- The impacts are local affecting only the MS-IP side of a MS-IP-MS call
- The implementation uses outband standard #7 signalling (a standard in the CCITT Signalling system) and has minor impacts on the Mobile Switching Centre (MSC) and the Base Station Subsystem (BSS)
- It can be used with all existing mobile voice codecs as long as they are supported by the IP network to which the gateway is connected
- Avoids the use of transceivers in Base Station Subsystem
- No need for speech coding and the use of Digital Signalling Processors (DSP) in the VoIP gateway

Tandem Free Operation (TFO) in GSM

In case of MS to MS calls in a mobile network without TFO, the speech signal is encoded within the first mobile station for transmission on the air interface, and transcoded within the first Tranceiver and Rate Adapter Unit (TRAU). The Pulse Code Modulation (PCM) samples are then transported within the fixed part of the network to the second TRAU using 64 kbit/s traffic links. The second TRAU encodes the speech signal a second time for the transmission on the second air interface. The two codecs of the connection are in "Tandem Operation".

This Tandem Operation has several disadvantages:

- The extra encoding/decoding degrades the speech quality more than necessary
- The links between the TRAUs need 64 kb/s while 16 or 8 kb/s would be sufficient

- The unnecessary encoding/decoding within the TRAUs allocates Digital Signalling Processor (DSP) power

The European Telecommunications Standards Institute (ETSI) is working on a standard for TFO, TS 04.53. This standard defines inband signals between TRAUs so that TFO effects only the TRAUs and are therefore fully compatible with existing equipment.

In FIG. 1 there is illustrated in a schematical manner, the principle of Tandem Free Operation (TFO), and in the following there will be give a brief description of this principle.

TFO in GSM is defined as an inband signalling protocol between two peer transceivers. It

- Tests the path for possible TFO
- Establishes the TFO connection
- Guarantees a fast fallback procedure
- Supports resolution of Codec mismatch

The standard defines both TFO Frames (speech) and TFO messages. TFO Frames affects only the TRAUs.

For Half Rate Speech Coding (HR) the required bandwidth is 8 kbits/s using the Least Significant Bit (LSB) of each PCM sample and for Full Rate Speech Coding (FR) and Enhanced Full Rate Speech Coding (EFR) 16 kbits/s, using the two LSB of each PCM sample.

The speech quality is of special concern for a Mobile Switching Centre (MSC) based gateway. This is because of the number of encodings/decodings that can occur for IP based mobile calls.

For normal mobile calls we have the two following scenarios (using GSM as an example):

MS -> PSTN:	GSM 06.10 - G.711
MS -> MS:	GSM 06.10 - G.711 - GSM 06.10

When IP is a part of a call leg the following can happen when GSM 06.10 is used as the IP audio codec:

MS -> IP -> PSTN:	GSM 06.10 - G.711 - GSM 06.1 - G.711
MS -> IP -> MS:	GSM 06.10 - G.711 - GSM 06.1 - G.711 - 06.10

Each encoding/decoding deteriorates the speech quality. For MS to MS calls we can hear the quality we get with two encodings. Adding one encoding as would be the case for an MS→IP→MS call will probably reduce the speech quality to an unacceptable level. One way to avoid this is by choosing G.711 as the IP codec, but then no reduction in bandwidth is achieved, which should be one of the main goals with using IP.

OBJECTS OF THE INVENTION

A main object of the present invention is to improve the speech quality, especially for mobile VOIP calls, by reducing the encoding/decoding to a minimum for thereby avoiding deterioration of speech quality.

Another object of the present invention is to also reduce the bandwidth, especially when using IP.

Another object of the present invention is to adapt the associated gateway (GW) so as to be in harmony with this optimisation.

BRIEF SUMMARY OF THE INVENTION

The above objects are achieved in an arrangement as stated in the preamble, which according to the present

invention is characterised by the features as stated in the enclosed patent claims.

In other words, the present invention suggests that the arrangement comprises means for either putting the TRAU in a transparent mode or letting the TRAU be bypassed altogether.

Further features and advantages of the present invention will appear from not only the enclosed patent claims, but also from the following description taken in conjunction with the enclosed drawings.

BRIEF DISCLOSURE OF THE DRAWINGS

FIG. 1 is a schematical diagram illustrating the principle of Tandem Free Operation (TFO).

FIG. 2 is a diagram illustrating the signalling sequences for TFO over IP.

DETAILED DESCRIPTION OF EMBODIMENTS

FIG. 1 is a schematical drawing illustrating the principle of Tandem Free Operation, and this principle has already been discussed on previous pages.

In FIG. 2 there is schematically illustrating signalling sequences for TFO over IP, these signalling sequences illustrating one of several embodiments wherein the general idea of the present invention has been implemented.

In the following there will be given a more detailed description of how this embodiment can be implemented.

TFO Over IP

The basic idea of the present invention is that no decoding is done in the TRAU. The TRAU can either be put in transparent mode or it is bypassed all together.

The MS encodes the speech either in Half Rate Speech Coding (HR), Full Rate Speech Coding (FR) or Enhanced Full Rate Speech Coding (EFR) and the speech samples are transmitted directly on to the IP network where they are assembled into Realtime Transfer Protocol (RTP)/User Datagram Protocol (UDP) packets. For HR the LSB bit on PCM is used, and for FR and EFR the two LSB bits are used.

The VoIP GW has to perform some Error Concealment. This is anyhow a normal function of a VoIP gateway.

Preferably this can be handled by standard #7 signalling using standard parameters, possibly using spare fields.

On the DTAP/BSSMAP side the BC (Bearer Capability) field is used, which is read by the Mobile Switching Centre (MSC) and mapped transparent to the User Service Information (USI) field on ISUP towards the VoIP Gateway. BC contains two fields for negotiation during call set-up. If the GW does not support the preferred BC, the default is applied.

The preferred field could contain "TFO wanted", and the default field "no TFO". In BC two spare bits in octet 3a could be applied.

VoIP gateway must have the final decision on which codec to use since it must terminate the correct TRAU frames when TFO is active.

For an outgoing call using TFO we could have the following scenario:

1. The user dials prefix to the destination number to indicate that he wants the call to be routed over an IP network.
2. After b-number analyses the MSC modifies the BC in SETUP. The BC contains two BC fields, one is fallback and one is preferred. The preferred BC is encoded with "TFO wanted", and codec type.
3. The MSC transfers the modified BC fields to the USI field in the outgoing IAM towards the VOIP gateway. The gateway now decides TFO or not and reads the codec type. The answer is transferred back to the MSC in the ACM message.
4. The MSC uses ASSIGNMENT REQUEST to request the wanted radio resources from the BSC. Information requiring TFO could be coded in fields like Channel Type, Classmark or others.
5. If TFO can be supported by the BSC this is confirmed back to the MSC in ASSIGNMENT COMPLETE. The BSC puts the TRAU in either transparent mode or bypassed mode. The method chosen here is of local (BSC) relevance only.
6. The MSC continues the call setup with the relevant DTAP messages.
7. If TFO was not accepted by the VOIP gateway the call is handled like a "no TFO" call.

Note:

In this scenario the codec type is decided by the gateway in case of TFO.

Fallback from TFO during speech is not considered necessary in a basic implementation.

For incoming TFO calls a reverse scenario applies. In case TFO is not supported by the BSC, or the MS does not support the incoming IP codec, the VOIP gateway must support fallback to G.711 speech.

Abbreviations

ACM	Address Complete Message
BSC	Base Switching Centre
BC	Bearer Capability
BSS	Base Station Subsystem
D-AMPS	Digital-Advanced Mobile Phone Service
DSP	Digital Signalling Processor
DTAP	Direct Transfer Application Part
ETSI	The European Telecommunications Standards Institute
EFR	Enhanced Full rate speech coding
FR	Full Rate speech coding
GSM	Global System for Mobile communication
GW	Gateway
HR	Half Rate speech coding
IAM	Initial Address Message
IP	Internet Protocol
ISUP	ISDN User Part
LSB	Least Significant Bit
MAP	Mobile Application Part
MS	Mobile Station
MSC	Mobile Switching Centre
PCM	Pulse Code Modulation
PDC	Personal Digital Cellular
PSTN	Public Switched Telephone Network
RTP	Realtime Transfer Protocol
SPE	Speech Encoding Eq
SPD	Speech Decoding Eq
TFO	Tandem Free Operation
TRAU	Tranceiver and Rate Adapter Unit
UDP	User Datagram Protocol
USI	User Service Information
VoIP	Voice over IP
ITU	(International Telecommunications Union, Geneva, Switzerland) formerly the CCITT (Consultative Committee for International Telephony and Telegraphy) is an international organization founded in 1865 and headquartered in Geneva that sets communications standards.

What is claimed is:

1. A method of improving speech quality for a call in a mobile Voice over IP (VoIP) network, said method comprising the steps of:

sending a first call setup-type message from an originating mobile station (MS) to a mobile switching center

(MSC), said call setup message including a bearer capability parameter;

performing a called number analysis by the MSC to determine a VoIP gateway for call routing;

determining by the MSC, whether Tandem Free Operation (TFO) is desired for the call;

sending a second call setup-type message from the MSC to the determined VoIP gateway, said second call setup-type message including a bearer capability field that is divided into a preferred field and a default field, the MSC setting the preferred field and inserting a codec type utilized by the originating MS when TFO is desired, and the MSC setting the default field when TFO is not desired;

determining in the VoIP gateway whether or not to utilize TFO;

sending an answer message from the VoIP gateway to the MSC indicating whether TFO is being utilized;

requesting by the MSC, required radio resources from a Base Station Controller (BSC) serving the originating MS, said BSC including a Transceiver and Rate Adapter Unit (TRAU) that transcodes an encoded speech signal from the originating MS if TFO is not being utilized;

notifying the BSC whether TFO is being utilized;

completing call setup;

transmitting the encoded speech signal from the originating MS to the BSC;

if TFO is not being utilized, transcoding the encoded speech signal from the originating MS in the TRAU prior to sending the signal from the BSC to the MSC, said transcoded speech signal being transmitted to the VoIP gateway where the signal is assembled into packets; and

if TFO is being utilized, bypassing the TRAU, and sending the encoded speech signal from the BSC to the MSC without transcoding, said encoded speech signal being transmitted to the VoIP gateway where the signal is assembled into packets.

2. The method of claim **1** wherein the step of transmitting the encoded speech signal from the originating MS to the BSC includes encoding the speech signal in the MS either in Half Rate Speech Coding (HR), Full Rate Speech Coding (FR), or Enhanced Full Rate Speech Coding (EFR), and wherein the encoded speech signal is assembled in the VoIP gateway into Real Time Transfer Protocol (RTP)/User Datagram Protocol (UDP) packets.

3. The method of claim **2** further comprising performing error concealment in the VoIP gateway utilizing spare fields in Signaling System 7 (SS7). message parameters.

* * * * *

US006529499B1

(12) **United States Patent**
Doshi et al.

(10) **Patent No.: US 6,529,499 B1**
(45) **Date of Patent: Mar. 4, 2003**

(54) **METHOD FOR PROVIDING QUALITY OF SERVICE FOR DELAY SENSITIVE TRAFFIC OVER IP NETWORKS**

(75) Inventors: **Bharat Tarachand Doshi**, Holmdel, NJ (US); **Enrique Hernandez-Valencia**, Highlands, NJ (US); **Kotikalapudi Sriram**, Marlboro, NJ (US); **Yung-Terng Wang**, Marlboro, MA (US); **On-Ching Yue**, Middletown, NJ (US)

(73) Assignee: **Lucent Technologies Inc.**, Murray Hill, NJ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/158,694**

(22) Filed: **Sep. 22, 1998**

(51) **Int. Cl.**[7] **H04L 12/66**; H04L 12/26

(52) **U.S. Cl.** **370/352**; 370/230; 370/468

(58) **Field of Search** 370/351–356, 370/400–402, 229–238.1, 522–524, 395.2, 395.21, 395.41, 395.52, 468; 709/223–229; 320/209

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,914,650 A		4/1990	Sriram	370/60
5,434,852 A	*	7/1995	La Porta et al.	370/524
5,463,620 A		10/1995	Sriram	370/60
5,732,078 A	*	3/1998	Arango	370/355
5,751,712 A	*	5/1998	Farwell et al.	370/431
6,064,653 A	*	5/2000	Farris	370/352
6,078,582 A	*	6/2000	Curry et al.	370/352
6,094,431 A	*	7/2000	Yamato et al.	370/395.21
6,097,722 A	*	8/2000	Graham et al.	370/395.21
6,205,211 B1	*	3/2001	Thomas et al.	379/125
6,292,478 B1	*	9/2001	Farris	370/352

FOREIGN PATENT DOCUMENTS

GB	2317 308 A	3/1998	H04L/12/46

OTHER PUBLICATIONS

"A New ATM Adaptation Layer for Small Packet Encapsulation and Multiplexing"; John H. Baldwin, Behram H. Bharucha, Bharat T. Doshi, Subrahmanyam Dravida and Sanjiv Nanda; Bell Labs Technical Journal, vol. 2, No. 2, Spring 1997.

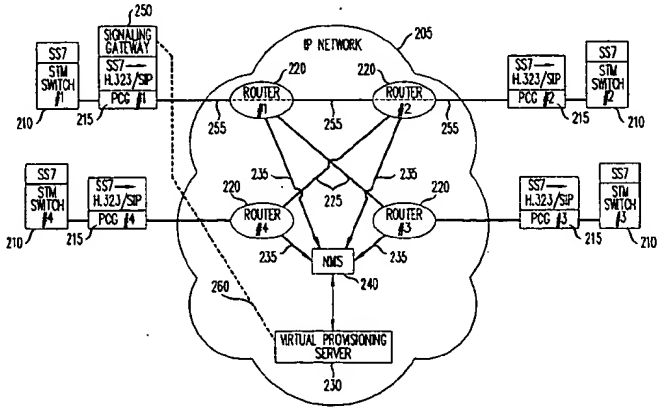
(List continued on next page.)

Primary Examiner—Steven Nguyen
(74) *Attorney, Agent, or Firm*—Troutman, Sanders, Mays & Valentine

(57) **ABSTRACT**

A quality of service guarantee for voice and other delay sensitive transmissions within an Internet Protocol (IP) network is provided by identifying the IP network path utilized for IP packet transmission between source and destination edge devices and virtually provisioning IP network path bandwidth for priority voice traffic. Priority for voice packets and admission control of new voice calls (and other delay sensitive traffic) based on the remaining available capacity over the IP network path guarantees that high priority voice (and other delay sensitive traffic) meet stringent delay requirements. A Virtual Provisioning Server is utilized to maintain bandwidth capacity data for each path segment within the IP network and to forward the bandwidth capacity data to a Signaling Gateway. The Signaling Gateway determines whether to accept or reject an additional delay sensitive traffic component based upon available bandwidth capacity for an IP network path. The Signaling Gateway then signals the originating source edge device as to its determination to accept or reject. Quality of Service guarantees concerning acceptable delay and jitter characteristics for real-time transmission over an IP network are therefore provided without the need to directly signal the individual IP routers over which an IP network path is established.

20 Claims, 5 Drawing Sheets



OTHER PUBLICATIONS

"Voice of ATM Using AAL 2 and Bit Dropping: Performance and Call Admission Control", Kotikalapudi Sriram and Yung-Terng Wang; Proceedings of the IEEE ATM Workshop, pp. 215–224, 1998.

"Anomalies Due to Delay and Loss in AAL2 Packet Voice Systems: Performance Models and Methods of Mitigation"; Kotikalapudi Sriram, Terry G. Lyons and Yung-Terng Wang; INFORMS Telecommun. Conf., Boca Raton, Fl. Mar. 8–11, 1998.

Kostas, T.J., et al., "Real-Time Voice Over Packet-Switched Networks", IEEE Network: The Magazine of Computer Communications, US, IEEE Inc., New York, vol. 12, No. 1, Jan. 1, 1998, pp. 18–27.

White, P.P. "RSVP and Integrated Services in the Internet: A Tutorial" IEEE Communications Magazine, US, IEEE Service Center, Piscataway, NJ, vol. 35, No. 5, May 1, 1997, pp. 100–106.

* cited by examiner

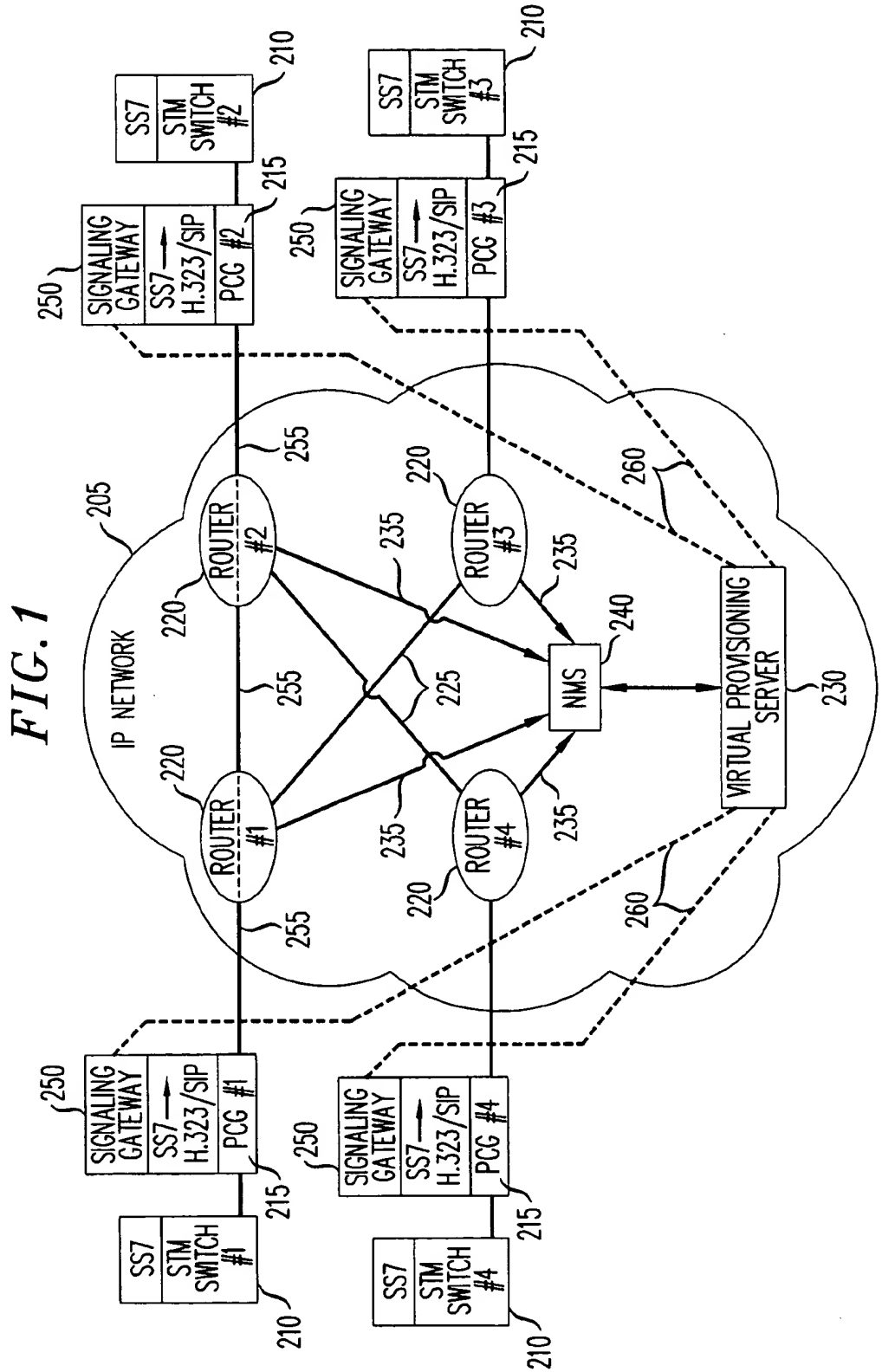

SS7
STM SWITCH #1
SS7
STM SWITCH #2
SS7
STM SWITCH #3
SS7
STM SWITCH #4
SIGNALING GATEWAY
SS7 → H.323/SIP
PCG #1
PCG #2
PCG #3
PCG #4
IP NETWORK
ROUTER #1
ROUTER #2
ROUTER #3
ROUTER #4
NMS
VIRTUAL PROVISIONING SERVER
205
210
215
220
225
230
235
240
250
255
260

FIG. 1

FIG. 2

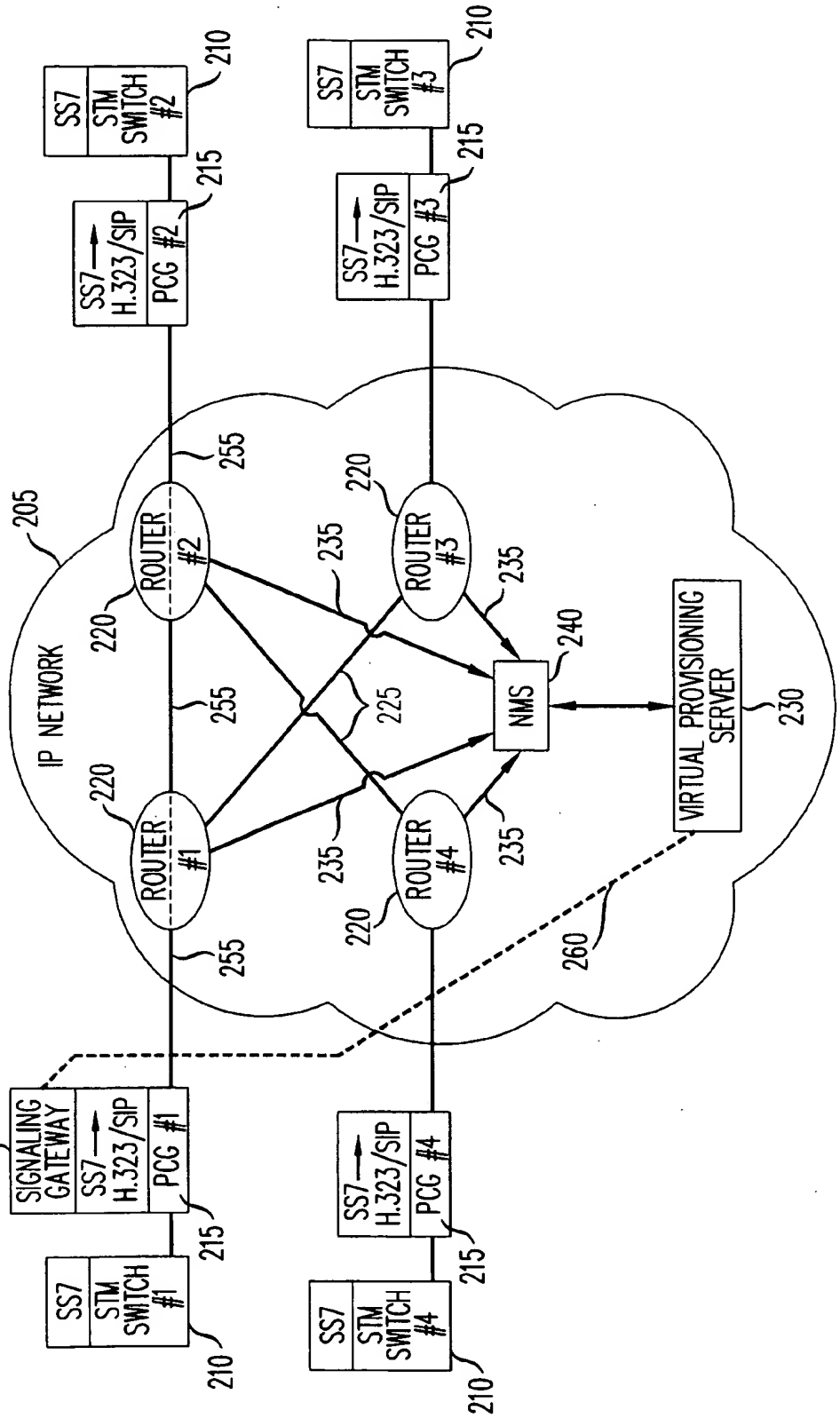

SS7 STM SWITCH #2
210
SS7 H.323/SIP PCG #2
215
SS7 STM SWITCH #3
210
SS7 H.323/SIP PCG #3
215
205
IP NETWORK
ROUTER #2
220
ROUTER #3
220
255
235
235
240
NMS
VIRTUAL PROVISIONING SERVER
230
225
ROUTER #1
220
ROUTER #4
220
255
235
235
260
250
SIGNALING GATEWAY
SS7 H.323/SIP PCG #1
215
SS7 H.323/SIP PCG #4
215
SS7 STM SWITCH #1
210
SS7 STM SWITCH #4
210

FIG. 3

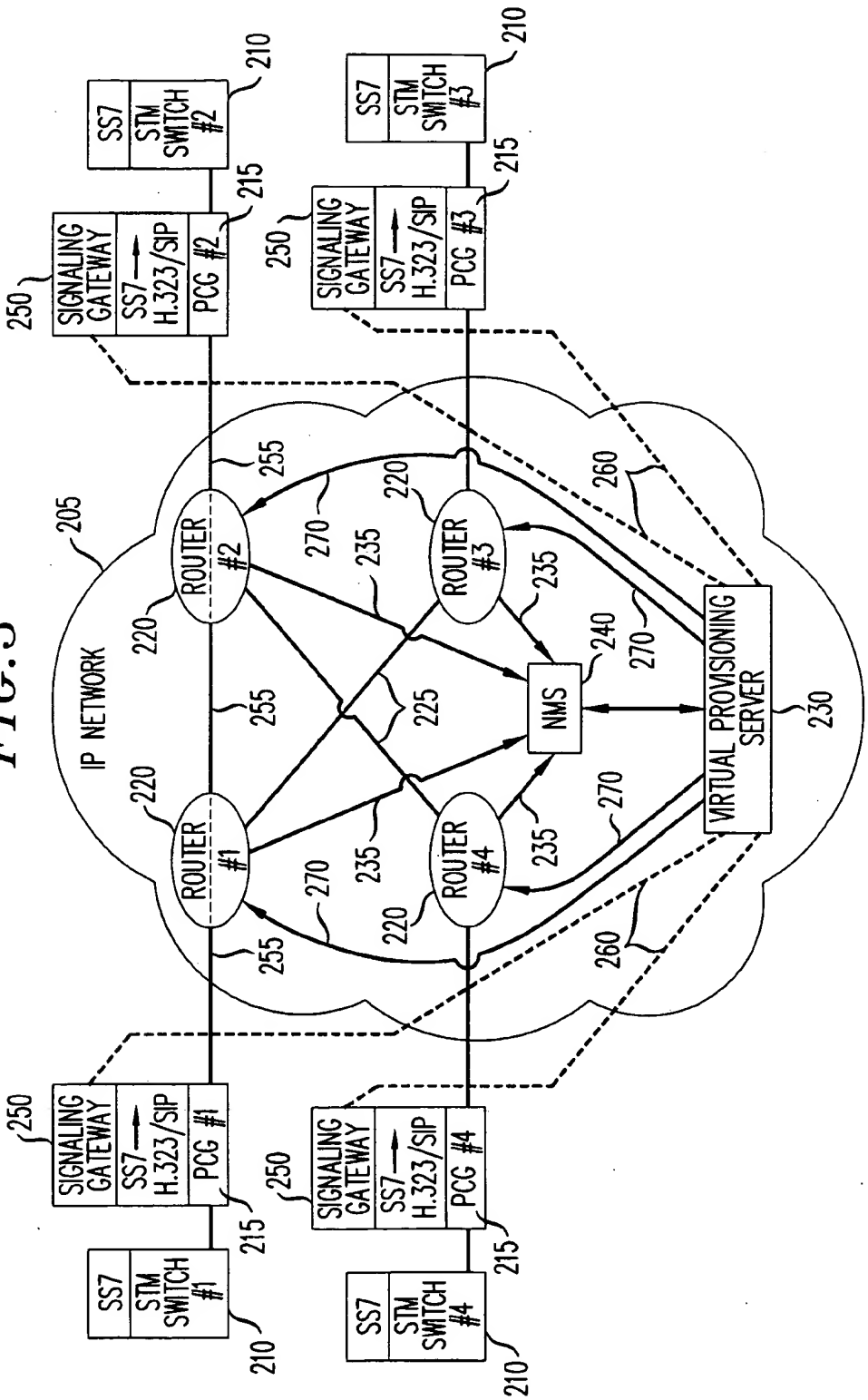

IP NETWORK
205
ROUTER #1
ROUTER #2
ROUTER #3
ROUTER #4
NMS
VIRTUAL PROVISIONING SERVER
230
SIGNALING GATEWAY
SS7 → H.323/SIP
PCG #1
PCG #2
PCG #3
PCG #4
SS7 STM SWITCH #1
SS7 STM SWITCH #2
SS7 STM SWITCH #3
SS7 STM SWITCH #4
210
215
220
225
235
240
250
255
260
270

FIG. 4

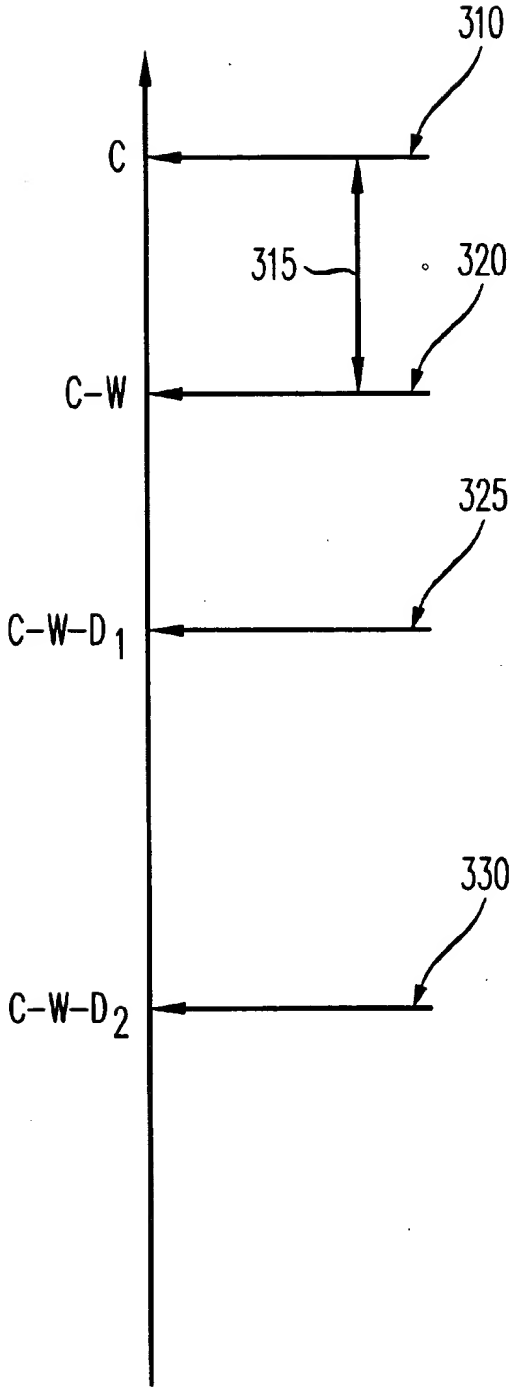
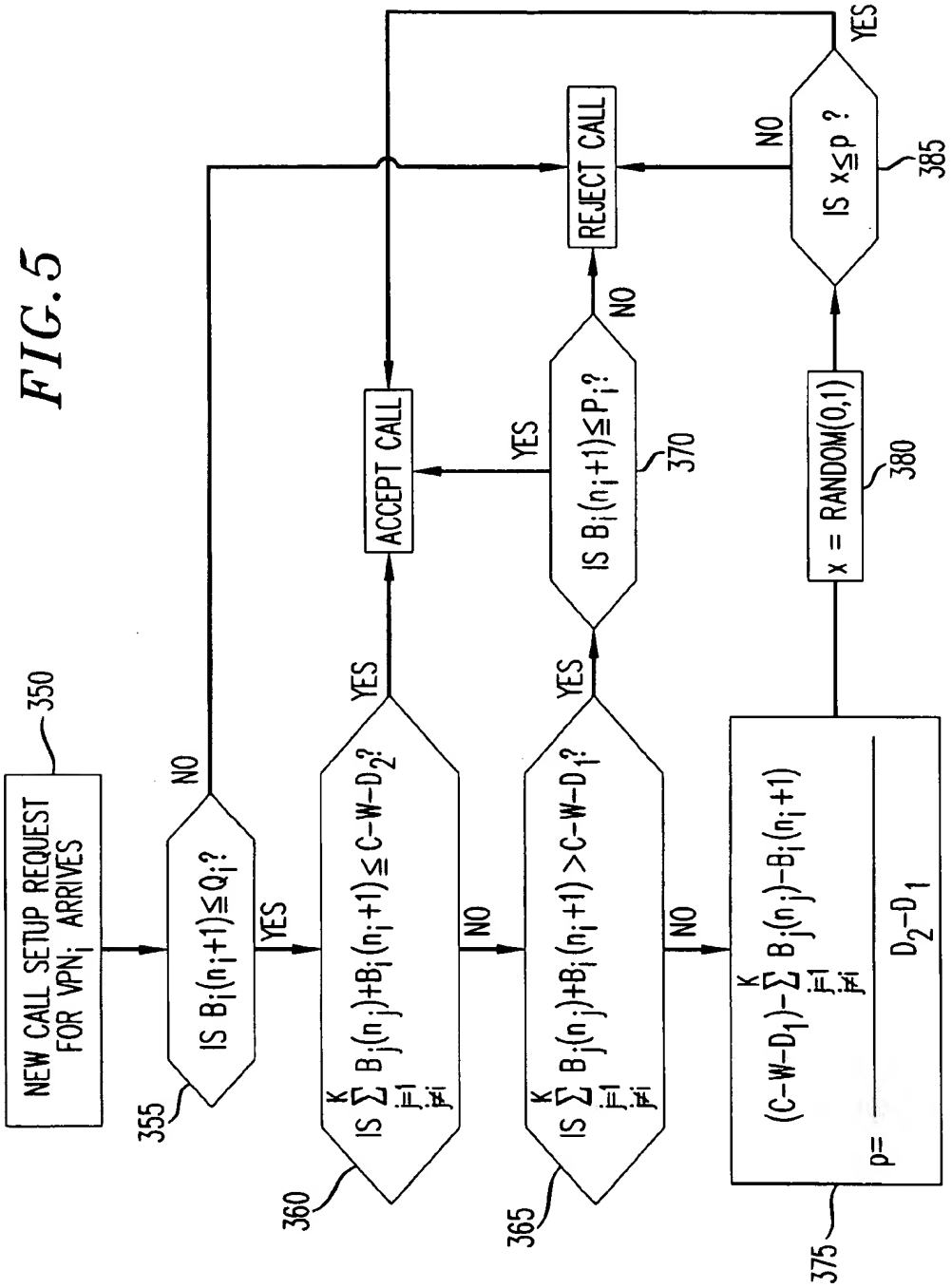


FIG. 5


NEW CALL SETUP REQUEST FOR VPN$_i$ ARRIVES
350
IS $B_i(n_i+1) \leq Q_i$?
355
NO
YES
IS $\sum_{j=1, j \neq i}^{K} B_j(n_j)+B_i(n_i+1) \leq C-W-D_2$?
360
YES
NO
ACCEPT CALL
IS $\sum_{j=1, j \neq i}^{K} B_j(n_j)+B_i(n_i+1) > C-W-D_1$?
365
YES
NO
IS $B_i(n_i+1) \leq P_i$?
370
YES
NO
REJECT CALL
$p = \frac{(C-W-D_1)-\sum_{j=1, j \neq i}^{K} B_j(n_j)-B_i(n_i+1)}{D_2-D_1}$
375
x = RANDOM(0,1)
380
IS $x \leq p$?
385
NO
YES

METHOD FOR PROVIDING QUALITY OF SERVICE FOR DELAY SENSITIVE TRAFFIC OVER IP NETWORKS

FIELD OF THE INVENTION

The present invention relates to the field of Internet Protocol (IP) networks, and more particularly to the transport of delay sensitive traffic over IP networks.

BACKGROUND OF THE INVENTION

A global network infrastructure for voice services, using a circuit-switching methodology, is supported by Public Switched Telephone and Private Branch Exchange networks. These networks utilize signaling to establish call connections and routing maps at network switches. The ability to signal during call connection set-up provides individual switches with the capability to reject call connection requests when that individual switch does not have the available bandwidth to support a new call connection. Since any switch in a connection path may reject a new call connection request based on available bandwidth limitations, switched voice networks are able to provide guaranteed Quality of Service to established connections. Quality of Service in switched voice networks is guaranteed because the governing precept is that it is preferable to block new call connection attempts rather than allow a new connected call to degrade the performance of established connected calls.

Explosive growth in Internet Protocol (IP) based Intranets and public Internet has generated a large network infrastructure of IP based routers. Recently, this large IP network infrastructure has begun to be utilized as a vehicle for real-time transmission of voice over the Internet, also known as Internet telephony. Each year, Internet telephony captures a greater share of the telephony market. However, unlike the case of switched voice service networks, routers contained within IP networks are not signaled. Since signaling between source, destination, and intermediate routers is not provided within IP networks, new calls can not be rejected at the IP routers, even if the routers are burdened beyond their respective bandwidth capacities. Therefore, real-time transmission over the Internet is subject to levels of delay and jitter not associated with Public Switched Telephone Networks and Private Branch Exchanges. Rather, transmission over the Internet and other IP networks is accomplished via a best effort transmission mode. Consequently, telephony over IP networks does not currently provide a Quality of Service guarantee for voice and other delay sensitive transmissions.

SUMMARY OF THE INVENTION

A quality of service guarantee for voice and other delay sensitive transmissions within an Internet Protocol (IP) network is provided by identifying the IP network path utilized for IP packet transmission between source and destination edge devices and virtually provisioning IP network path bandwidth for priority voice traffic. Priority for voice packets and admission control of new voice calls (and other delay sensitive traffic) based on the remaining available capacity over the IP network path guarantees that high priority voice (and other delay sensitive traffic) meet stringent delay requirements. A Virtual Provisioning Server is utilized to maintain bandwidth capacity data for each path segment within the IP network and to forward the bandwidth capacity data to a Signaling Gateway. The Signaling Gateway determines whether to accept or reject an additional delay sensitive traffic component based upon available bandwidth capacity for an IP network path. The Signaling Gateway then signals the originating source edge device as to its determination to accept or reject. Quality of Service guarantees concerning acceptable delay and jitter characteristics for real-time transmission over an IP network are therefore provided without the need to directly signal the individual IP routers over which an IP network path is established.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the present invention may be obtained from consideration of the following description in conjunction with the drawings in which:

FIG. **1** is a diagram illustrating a voice over IP network between Packet Circuit Gateway edge devices and incorporating a Virtual Provisioning Server, the Virtual Provisioning Server communicating with a plurality of Signaling Gateways, in accordance with an exemplary embodiment of the present invention;

FIG. **2** is a diagram illustrating a voice over IP network between Packet Circuit Gateway edge devices and incorporating a Virtual Provisioning Server, the Virtual Provisioning Server communicating with a Signaling Gateway co-located with one Packet Circuit Gateway, and providing Signaling Gateway functionality to more than one Packet Circuit Gateway **215** within the network, in accordance with an exemplary embodiment of the present invention;

FIG. **3** is a diagram illustrating a voice over IP network between Packet Circuit Gateway edge devices and incorporating a Virtual Provisioning Server, the Virtual Provisioning Server further performing functions as a Virtual Private Network (VPN) Resource Manager, in accordance with an exemplary embodiment of the present invention;

FIG. **4** is a diagram illustrating the bandwidth allocation structure associated with an exemplary embodiment of the present invention; and

FIG. **5** is a flow diagram illustrating one exemplary embodiment of an algorithm for call admission control for a plurality of Virtual Private Networks sharing a link within a common network, in accordance with the present invention.

DETAILED DESCRIPTION

FIGS. **1**, **2**, and **3** are diagrams illustrating various embodiments for IP networks **205** between Packet Circuit Gateway edge devices **215** incorporating a Virtual Provisioning Server **230**, in accordance with the present invention. In FIG. 1, the Virtual Provisioning Server **230** communicates with a Signaling Gateway **250** associated with each Packet Circuit Gateway edge device **215**. In FIG. 2, the Virtual Provisioning Server **230** communicates with a Signaling Gateway **250** co-located with one Packet Circuit Gateway **215**, and providing Signaling Gateway functionality to more than one Packet Circuit Gateway **215** within the network. In FIG. 3, the Virtual Provisioning Server **230** performs additional functions as a Virtual Private Network Resource Manager.

The present invention is described as being utilized within an environment wherein voice traffic originates and terminates on regular Public Switched Telephone Network circuit switches, such as Synchronous Transfer Mode switches **210**, and is carried over paths between routers within an IP network **205**. However, these circuit switches may also be implemented as simple access multiplexers or edge vehicles as would be apparent to those skilled in the art. It would also

be apparent to those skilled in the art that the present invention may be practiced with any IP datagram traffic (in addition to voice), although the present invention provides the greatest benefit for the transport of delay sensitive IP datagram traffic. Conversion from a circuit signal to IP format occurs at Packet Circuit Gateways (PCGs) **215**, which are also alternatively known as Service Access Concentrators (SACs) or Internet Telephone Gateways. In addition to conversion between circuit and IP formats, Packet Circuit Gateways **215** also provide voice compression/decompression, silence suppression/insertion, and other well known functions needed for specific applications.

Signaling Gateways **250** are utilized to provide the appropriate interface and interworking between signaling mechanisms and also to determine acceptance or rejection of a new call request originating from an associated Packet Circuit Gateway. Circuit networks, such as Public Switched Telephone Networks, typically use Signaling System 7 (SS7) to communicate requests for connection set-up and tear down. IP endpoints and intermediate routers use ITU-T H.323 or Session Initiation Protocol (SIP) for session management. Therefore, Signaling Gateways **250** provide a higher layer protocol utilized at the Packet Circuit Gateways **215** to facilitate conversions in signaling mechanisms between Public Switched Telephone Networks and IP networks **205**. It should be noted that a resident Signaling Gateway **250** is not required at each Packet Circuit Gateway Rather, the Signaling Gateway function may be implemented at a single location for all Packet Circuit Gateways with control signals transmitted to corresponding Packet Circuit Gateways from the single Signaling Gateway. For example, FIGS. **1** and **3** illustrate embodiments of the present invention wherein each Packet Circuit Gateway **215** maintains a resident Signaling Gateway **250**. However, FIG. **2** illustrates an embodiment of the present invention wherein only PCG **#1** maintains a resident Signaling Gateway **250**. The Signaling Gateway functions are provided at PCG**#2**, PCG**#3**, and PCG**#4** by transmission of appropriate control signals between the Signaling Gateway resident at PCG**#1** and the remaining Packet Circuit Gateways. Transmission may be over the serviced IP network **205** within a TCP/IP session, an adjunct transmission medium, or any other well known means for data transport.

One unique feature of the present invention is provided by a Virtual Provisioning Server **230**. The Virtual Provisioning Server is utilized to provide the Signaling Gateways **250** with network bandwidth capability information, so that the Signaling Gateways are able to make a determination as to whether to accept or reject a new call request at an associated Packet Circuit Gateway **215**. The basis for admission/denial decisions for new calls is made in order to provide assurances that Quality of Service characteristics, such as delay, jitter, and loss of call connections, are maintained below a guaranteed threshold for established voice call connections.

The Virtual Provisioning Server **230** communicates the network bandwidth capability information to the Signaling Gateways **250** at least once at the commencement of network operation, and episodically whenever the underlying IP network is subject to changes to its link bandwidths due to link failures, new link establishment, addition of bandwidth to existing links, etc. A Network Management System (NMS) is typically associated with an IP network and its functions well known in the art. However, in association with the present invention the Network Management System performs the additional function of apprising the Virtual Provisioning Server of any changes to the link bandwidths as enunciated above.

FIGS. **1–3** illustrate a network path **255** for the transport of IP packets between PCG**#1** and PCG**#2**. The path **255** is via intermediate components Router **#1** and Router **#2**. Routers **220** are interconnected at the physical layer within the IP network **205** by a plurality of physical layer router transport segments **225**. It is over a plurality of these physical layer router transport segments **225** that the illustrated network path **255** is established. A network path **255** is comprised of a plurality of path links established over the plurality of physical layer router transport segments **225**. The Virtual Provisioning Server **230**, in cooperation with the Public Switched Telephone Network provisioning mechanism and admission control implemented by the Signaling Gateway **250**, provides for a quality guarantee to voice traffic while allowing the remaining capacity in the IP network to be used by other traffic utilizing the well known best effort mode. Similar provisioning can extend the service guarantee to multiple classes of traffic, for example—video conferencing.

Given that specific STM switches **210** are tied to corresponding Packet Circuit Gateways **215**, voice call transport capacity can be easily predicted using standard traffic engineering methods to determine the capacity needed between Packet Circuit Gateways **215**. Specific format variables, such as the type of compression method used, the silence suppression capability, etc., determine the network path bandwidth requirements between each pair of Packet Circuit Gateways **215**. The Virtual Provisioning Server **230** maintains and manages data corresponding to the transmission capacities of the IP network routers **220** and the physical layer router transport segments **225** between those routers **220**. The Virtual Provisioning Server is used, in accordance with the present invention, to determine the capacity requirements over each path between IP network routers **220** to meet the needed bandwidth requirements between Packet Circuit Gateways **215**. The capacity requirements over each network element, such as routers **220** and physical layer router transport segments **225** are virtually provisioned within available bandwidth capacity for delay sensitive traffic requirements. In accordance with the present invention, the bandwidth is considered virtually provisioned since the admission/denial of new connected calls is not controlled at each individual router **220**, but rather at the Packet Circuit Gateway edge devices **215**. Remaining bandwidth capacity over network elements is made available to delay insensitive packet transport only after the provisioning of bandwidth for delay sensitive voice frames or IP packets at the Packet Circuit Gateways **215** is performed. Alternatively, a provisioned minimum bandwidth capacity over each IP network path may be reserved for delay insensitive traffic, with the remaining bandwidth allocated for use by delay sensitive traffic. A Type-of-Service (TOS) field in the IP packet header is utilized to distinguish between delay sensitive and delay tolerant traffic types. Thus, voice packets may be given priority over data packets to ensure that delay and packet loss is in accordance with Quality of Service requirements.

If IP network routers **220** and physical layer router transport segments **225** utilized for a specific path **255** do not have the necessary bandwidth capacity to meet determined capacity requirements, the Virtual Provisioning Server **230** allocates portions of the bottleneck capacity to the pairs of Packet Circuit Gateways **215** competing for this capacity and messages the associated Signaling Gateway **250** of this allocation. The Virtual Provisioning Server **230** also calculates the need for added capacity within the IP network **205** to meet current and future bandwidth needs. By centrally

calculating and determining required network bandwidth provisioning and messaging the Signaling Gateways **205** within the IP network **205** of the bandwidth allocation, the Virtual Provisioning Server **230** determines the maximum number of voice calls that can be supported simultaneously between any pair of Packet Circuit Gateways **215**. Since Signaling Gateways **250** provide the signaling interworking between SS7 and H.323/SIP, they are also able to track the number of connected calls in progress between pairs of Packet Circuit Gateways **215**. As shown in the embodiment of the present invention illustrated in FIG. **2**, and as previously described, one Signaling Gateway **250** may be utilized to control more than one Packet Circuit Gateway **215** and may also be utilized to track the number of connected calls in progress between other network Packet Circuit Gateways **215** (PCG #**2**, PCG #**3**, and PCG #**4** in the instant embodiment as illustrated in FIG. **2**).

As previously described, the Virtual Provisioning Server **230** also exchanges data with a Network Management System (NMS) **240**. The Network Management System is a well known network controller used to maintain IP network **205** information pertaining to network element capacities, network bandwidth and capacity demand and growth data, link failures, etc. The Network Management System **240** is operable to exchange messages and signals with network routers **220** and to provide and maintain this network information via signaling channels **235**. However, the Network Management System **240** does not determine or control admission/denial decisions for new call connections at the Packet Circuit Gateways **215**. The Network Management System **240** provides the Virtual Provisioning Server **230** with information about the IP network **205** topology, capacities, failure events, etc. The Virtual Provisioning Server **230** uses this information to update its calculations and signals the Network Management System **240** if changes need to be implemented within the IP network, such as updating routing algorithm weights. Routing algorithm weights are used to determine the routing path for forwarding an IP packet. The use and implementation of such routing algorithm weights is well known in the art of IP networking. When needed capacities cannot be achieved temporarily due to failure events, the Virtual Provisioning Server **230** determines the maximum number of calls that can be supported on affected paths throughout the network and informs the associated Signaling Gateways **250**, thereby providing a mechanism to throttle the number of connected calls at the various network Packet Circuit Gateway edge devices **215**.

Although the instant embodiment of the present invention is described in the context of connectivity between PSTN switches and Signaling Gateways **250** to manage signaling conversion and admission control, it may also be used to support telephony between PCs and telephony between a PC and a phone via a PSTN switch. In order to guarantee connection quality for these connections, it is important to provide messaging from the Virtual Provisioning Server **230** to the Signaling Gateway **250**, thus informing the Signaling Gateway about the call capacities for PCG-to-PCG paths for a minimum of telephony traffic originating from PSTN and PCs. In addition, since a network operator may not control the coding rate in this case (i.e.—when calls originate from PCs), a traffic policing function is utilized at the PCG to monitor compliance with the traffic assumptions used in call set-up signaling.

Voice calls originating from a PC may be assigned lower priority as compared to those originating from a PSTN. Doing so allows the Signaling Gateway **250** to reject PC originated calls based on a lower bandwidth utilization, and rejects the PSTN originated calls at a higher threshold. Therefore, the Signaling Gateway **250** can guarantee call connection quality for voice and other Quality of Service sensitive services by enforcing call admission control at the Packet Circuit Gateways **230** and preferentially awarding priority for PSTN originated voice services over other services. In addition, a service provider may provide a plurality of critical service guarantees to customers and similarly, multiple customers may desire similar critical service guarantees over common paths within an IP network **205**. One such example is presented within the context of Virtual Private Networks for voice traffic, wherein a network provider provides wide area services to interconnect corporate users in different locations. The ability to provide multiple Virtual Private Networks along with public service over a common infrastructure is attractive to both the service provider and corporate customers. One critical benefit of providing a Virtual Private Network is that the service provider is able to deliver secure access to the user. A second benefit is the ability to provide a Quality of Service guarantee comparable to that on leased private lines between customer premises switches (e.g., PBXs).

Virtual Private Network customers negotiate bandwidth and service quality guarantees from a wide area network operator or service provider. The network operator guarantees this negotiated service level to all Virtual Private Network customers by utilizing the common infrastructure to achieve multiplexing gain. Capabilities available in currently available routers **220** allow the Virtual Provisioning Server **230** to provide these guaranteed services. For example, routers are available which are capable of identifying flows based on the port, source, and destination identifiers, and which categorize group flows into classes and/or super classes according to the level of service and bandwidth guarantees negotiated. These routers are also operable to allocate and manage minimum and maximum bandwidth for each class, super class, etc. Incorporation of buffer and queue management at the routers provides distinction and differentiation of priority treatment among flow classes and super classes. Additionally, statistical multiplexing may be provided for flows within a class and/or among classes within a super class. A system of Weighted Fair Queuing (WFQ) service provides for management of flow, class, and super class bandwidths. If one of the classes or super classes exceeds a negotiated bandwidth allocation, superior service quality may still be provided if the other negotiated classes or super classes are not completely utilizing their allocated bandwidth. Therefore, only the Quality of Service provided to classes or super classes exceeding their negotiated allocation of bandwidth are affected.

Referring to FIG. **3**, the Virtual Provisioning Server **230** is utilized as a Virtual Private Network Resource Manager. The Virtual Private Network Resource Manager utilizes optimizing algorithms to (i) partition bandwidth between Virtual Private Networks and within Virtual Private Networks if the customer desires a further subclassification of services and (ii) control flow routing within the network. If the network routers **220** utilized have flow partitioning capability, but do not have flexible routing capability, then flow routes are fixed through the IP network **205** and capacities are partitioned in the network by the Virtual Private Network Resource Manager based upon the negotiated Virtual Private Network contract. The Virtual Provisioning Server **230**, functioning as a Virtual Private Network Resource Manager, sends this partitioning information to individual routers **220** within the network **205** so that the

network routers **220** are able to set algorithm weights, minimum bandwidth, maximum bandwidth, buffer thresholds, etc. Communication between the Virtual Private Network Resource Manager is illustrated over a VPN signaling path **270** between the Virtual Provisioning Server **230** and individual routers, in accordance with FIG. **3**. The illustrated VPN signaling path **270** is merely illustrative, and any number of other means for signaling routers **220** would also be apparent to those skilled in the art, including communicating through the Network Management System **240**. Once partitioning information is received at network routers **220** and partitioning is accomplished, each Virtual Private Network is established with its allocated minimum bandwidth.

Referring again to FIGS. **1–3**, Virtual Private Networks for voice may also be supported using PSTN switches or multiplexers as access vehicles (STM switches **210** in the instant example) and utilizing the IP network **205** as backbone, as was previously described. Advantageously, the instant embodiment for establishing Virtual Private Networks for voice is achieved using network routers **220** with simple priority mechanisms. That is, signaling is not required between the Virtual Provisioning Server **230** and network routers **220** to establish and maintain the Virtual Private Networks. Rather, the Virtual Provisioning Server **230** uses aggregate capacity needed between a pair of gateways to perform virtual provisioning. The Packet Circuit Gateways **215**, in conjunction with the Signaling Gateways **250**, are utilized to control the acceptance or rejection of new calls from each Virtual Private Network customer utilizing an acceptance/rejection algorithm residing in the Virtual Provisioning Server **230**.

FIGS. **4** and **5** illustrate and define an exemplary algorithm for performance of the acceptance or rejection of new calls over a Virtual Private Network established between Packet Circuit Gateways **215**, in accordance with the present invention. In conjunction with the accompanying description, the following definitions apply:

C=The total link bandwidth **310**,

W=The minimum bandwidth always available for combined traffic supported using Available Bit Rate (ABR) or best effort data service **315**,

C–W=The total bandwidth available for call admission control purposes **320**,

$C-W-D_1$=An upper threshold for call admission control purpose **325**,

$C-W-D_2$=A lower threshold for call admission control purpose **330**,

$B_i(n_i)$=Bandwidth needed to support n_i connections for VPN_i with a specified Quality of Service,

P_i=Minimum bandwidth contracted for VPN_i,

Q_i=Maximum bandwidth contracted for VPN_i, and

K=Number of Virtual Private Networks with Quality of Service guarantees sharing the link in consideration.

When a new call set-up request for VPN_i arrives at the Signaling Gateway **250**, then the exemplary algorithm associated with FIG. **5** is performed to determine whether to accept or reject the new call, in accordance with step **350**. The bandwidth utilized by K Virtual Private Networks (VPN_i; i=1,2,3, . . . K) is monitored at the Signaling Gateway **250**. Referring to step **355**, when the VPN_i bandwidth necessary to support an additional call exceeds the maximum bandwidth allocation (Q_i), the requested new call is rejected. However, when the VPN_i bandwidth necessary to support an additional call does not exceed the maximum bandwidth allocation (Q_i), then step **360** is performed. In accordance with step **360**, if the VPN_i bandwidth usage would be between the range of zero to $(C-W-D_2)$ after connecting the new call, then the new call is accepted. However, if VPN_i bandwidth usage would be greater than $(C-W-D_2)$ after connecting the new call, then step **365** is performed. In accordance with step **365**, if VPN_i bandwidth usage would be between the range from $(C-W-D_1)$ to (C–V), a new call set-up request for VPN_i is accepted only if the bandwidth usage by VPN_i has not exceeded its minimum allocation, P_i, otherwise the call is rejected, in accordance with step **370**. If however, the VPN_i bandwidth usage is between the range of $(C-W-D_2)$ to $(C-W-D_1)$, a new call set-up request for VPN_i is accepted or rejected probabilistically based on a sliding scale algorithm in accordance with step **375**. Let q=(1–p) denote the ratio of bandwidth usage in excess of $(C-W-D_2)$ over (D_2-D_1). A random number x is generated at the Signaling Gateway **250** to support the probabilistically based algorithm, in accordance with step **380**. If the value of x is less than or equal to probability p, then the new call is accepted, in accordance with step **385**. For a call that traverses multiple links between its source and destination PCGs, the algorithm of FIG. **4** and FIG. **5** is repeated for each path link used to establish the call. The call is connected between the source and destination PCGs only if the algorithm yields a positive determination (to accept the call) for each link in the path.

During implementation of the exemplary algorithm of FIG. **5**, the bandwidth utilization data, $B_i(n_i)$, as a function of the number, n_i, for calls over VPN_i is utilized. If the calls or connections are constant bit rate, then $B_i(n_i)$ is a simple linear function of n_i. However, if the calls or connections are variable bit rate by nature or by design, for example—voice with silence elimination, on/off data sources, etc., then $B_i(n_i)$ is typically a non-linear function of n_i. The non-linear nature of $B_i(n_i)$ is due to the statistical multiplexing of randomly varying variable bit rate sources, as is well known in the art. For example, the specific nature of a $B_i(n_i)$ function, in the context of packet voice multiplexing, is detailed in a publication by K. Sriram and Y. T. Wang entitled "Voice Over ATM Using AAL2 and Bit Dropping: Performance and Call Admission Control," Proceedings of the IEEE ATM Workshop, May 1998, pp. 215–224, which is incorporated herein by reference.

Prior reference to the Virtual Provisioning Server (VPS) is described in the context of an IP network which includes multiple interconnected Open Shortest Path First (OSPF) domains. The present invention may also be implemented within an IP network comprised of multiple interconnected administrative areas, wherein each administrative area is comprised of multiple OSPF domains. Typically, each administrative area is an IP network belonging to an individual internet service provider or carrier, although such a configuration is not required. Such an embodiment of the present invention may be implemented with each administrative area having one gateway VPS. Each respective VPS may be co-located with the gateway router for that respective administrative area, although co-location is not a required aspect of the embodiment. Each pair of respective gateway VPSs determines the capacity requirements between their respective gateway routers. Further, each gateway VPS provides the necessary bandwidth capacity information between pairs of neighboring administrative areas to the VPSs located in each of the OSPF domains within its administrative area. Thus, the signaling gateways anywhere in the larger IP network are adequately provided with the necessary information for admission/denial of calls,

including those that originate in one administrative area and terminate in another.

Numerous modifications and alternative embodiments of the invention will be apparent to those skilled in the art in view of the foregoing description. For example, although the present invention has been described in the context of a single Virtual Provisioning Server utilized to service an entire IP network and control all Signaling Gateways within that network, it is also equally applicable for an embodiment of the present invention operable for multi-domain operation. That is, for those instances when call routing is made from a first telephony gateway source connected to a first IP domain and the destination is a second telephony gateway connected through another IP domain, the call processing involves intra-domain routing to the gateway router in the first domain, routing among gateway routers in intervening domains, and intra-domain routing from the gateway router to the telephony gateway in the last domain. Protocols such as Open Shortest Path First (OSPF) determine routing in a domain while a Border Gateway Protocol (BGP) is used for inter-domain routing between gateway domains. In such an embodiment of the present invention, a plurality of Virtual Provisioning Servers are utilized, one for each IP domain. Each Virtual Provisioning Server manages the virtual provisioning of routers within its respective domain, including Gateway Border Routers. Additionally, each pair of interfacing Virtual Provisioning Servers determines the capacity requirements between their respective pair of interfacing Gateway Border Routers. As was true for the single domain embodiment of the present invention, admission/denial control at the originating and terminating Packet Circuit Gateways is enabled without signaling the incorporated routers directly. In the multi-domain embodiment, this capability is attributable to shared knowledge of intra-domain and inter-domain routing protocols among the interfaced Virtual Provisioning Servers and also due to the static nature of router algorithm weights.

Additionally, the previous description is applicable for embodiments of the present invention in which service guarantees are provided without adding signaling mechanisms between routers and the associated Virtual Provisioning Server. However, the present invention would be equally applicable for those instances in which the Virtual Provisioning Server is operable to directly signal the network routers; although such an embodiment would be more accurately described as having a Server in which the provisioning is more real than virtual (since the provisioning is controlled at the routers instead of at the corresponding originating and terminating gateways). This alternative embodiment utilizes state exchange protocols in Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP), which are extended to provide dynamic topology and capacity information.

The present invention may also be used in evolving IP networks in which the well-known Multi-Protocol Label Switching (MPLS) is utilized at the network IP routers. In an MPLS based IP network, the Virtual Provisioning Server maintains a knowledge base of possible multiple paths between source-destination pairs of Packet Circuit Gateway edge devices. The Signaling Gateways receive information from the Virtual Provisioning Server about alternative paths and associated capacities between PCG pairs, and admits a new voice call request if capacity is available over any of the available paths, otherwise, the call request is rejected.

Accordingly, this description is to be construed as illustrative only and is for the purpose of teaching those skilled in the art the best mode of carrying out the invention and is not intended to illustrate all possible forms thereof. It is also understood that the words used are words of description, rather than limitation, and that details of the structure may be varied substantially without departing from the spirit of the invention and the exclusive use of all modifications which come within the scope of the appended claims are reserved.

What is claimed is:

1. A method for providing a Quality of Service guarantee for delay sensitive traffic conveyed over a path within an Internet Protocol (IP) network having a virtual provisioning server, a source edge device providing an interface for launching said delay sensitive traffic within said IP network, said method comprising the steps of:

receiving, at a signaling gateway, a value representing a bandwidth capacity for said path;

receiving at said signaling gateway, a request to establish an additional delay sensitive traffic component over said path;

comparing, at said signaling gateway, said value representing said bandwidth capacity for said path with a total bandwidth needed if said additional delay sensitive traffic component is established over said path;

identifying, at said signaling gateway, at least one of a plurality of paths within said IP network as having a most limiting available bandwidth capacity, wherein the identified path has sufficient available bandwidth capacity to handle said additional delay sensitive traffic component; and

limiting said quantity of said delay sensitive traffic launched from said source edge device to less than or equal to said most limiting available bandwidth capacity.

2. The method in accordance with claim **1** wherein said value representing said bandwidth capacity for said path is transmitted from said virtual provisioning server to said signaling gateway.

3. The method in accordance with claim **1** wherein said request to establish said additional delay sensitive traffic component over said path is conveyed from said source edge device.

4. The method in accordance with claim **3** wherein said source edge device is a packet circuit gateway.

5. The method in accordance with claim **1** further comprising the step of conveying said signal denying said request to establish said additional delay sensitive traffic component from said signaling gateway to said source edge device.

6. The method in accordance with claim **1** further comprising the steps of:

generating, at said signaling gateway, a signal authorizing said request to establish said additional delay sensitive traffic component if said total bandwidth needed is less than or equal to said value representing said bandwidth capacity for said path; and

conveying said signal authorizing said request to establish said additional delay sensitive traffic component from said signaling gateway to said source edge device.

7. The method in accordance with claim 1, further comprising:

generating, at said signaling gateway, a signal denying said request to establish said additional delay sensitive traffic component if said total bandwidth needed is greater than said value representing said bandwidth capacity for said path.

8. A method for providing a Quality of Service guarantee for real-time voice transmission traffic conveyed between a

source Packet Circuit Gateway and a destination Packet Circuit Gateway over an Internet Protocol (IP) network having a plurality of routers, said source Packet Circuit Gateway providing an interface for launching said real-time voice transmission traffic within said IP network over an IP network path, said method comprising the steps of:

partitioning, from a bandwidth capacity associated with said IP network path, a first provisioned bandwidth capacity for a first Virtual Private Network (VPN), said VPN contracted for said real-time voice transmission traffic conveyed between said source Packet Circuit Gateway and said destination Packet Circuit Gateway;

maintaining, at a Signaling Gateway, a value representing said first provisioned bandwidth capacity for said first VPN;

receiving, at said Signaling Gateway, a request from said source Packet Circuit Gateway to establish a new call connection with said destination Packet Circuit Gateway over said first VPN, in addition to a plurality of presently established call connections;

comparing, at said Signaling Gateway, said value representing said first provisioned bandwidth capacity for said first VPN with a required first VPN bandwidth capacity should said new call connection be established; and

transmitting, from said Signaling Gateway, a signal denying said request to establish said new call connection if said required first VPN bandwidth capacity should said new call connection be established is greater than said value representing said first provisioned bandwidth capacity for said first VPN.

9. The method in accordance with claim **8** further comprising the step of:

transmitting, from said Signaling Gateway, a signal authorizing said request to establish said new call connection if said required first VPN bandwidth capacity should said new call connection be established is less than or equal to said value representing said first provisioned bandwidth capacity for said first VPN.

10. The method in accordance with claim **8** wherein a Virtual Provisioning Server is utilized to provide said Signaling Gateway with said value representing said first provisioned bandwidth capacity for said first VPN.

11. The method in accordance with claim **10** wherein said Virtual Provisioning Server is adapted to maintain a plurality of Virtual Private Networks over said IP network path.

12. The method in accordance with claim **8** wherein said Quality of Service guarantee is established by maintaining delay of said real-time voice transmission traffic conveyed between said source Packet Circuit Gateway and said destination Packet Circuit Gateway below a guaranteed threshold value.

13. The method in accordance with claim **8** wherein said Quality of Service guarantee is established by maintaining jitter of said real-time voice transmission traffic conveyed between said source Packet Circuit Gateway and said destination Packet Circuit Gateway below a guaranteed threshold value.

14. The method in accordance with claim **8** wherein a circuit network switch is utilized to supply and accept said plurality of presently established call connections and said new call connection from said source Packet Circuit Gateway.

15. The method in accordance with claim **14** wherein said circuit network switch is a Synchronous Transfer Mode (STM) switch.

16. The method in accordance with claim **8** wherein at least one of said plurality of routers is operable to support Multi-Protocol Label Switching.

17. The method in accordance with claim **10** wherein a plurality of Multi-Protocol Label Switching (MPLS) routers is utilized to establish a plurality of paths between said source Packet Circuit Gateway and said destination Packet Circuit Gateway.

18. The method in accordance with claim **17** wherein said Virtual Provisioning Server is further operable to provide said Signaling Gateway with a plurality of values representing bandwidth capacities for each of said plurality of paths between said source Packet Circuit Gateway and said destination Packet Circuit Gateway.

19. The method in accordance with claim **10** wherein a plurality of Virtual Provisioning Servers are utilized to service a corresponding plurality of Open Shortest Path First domains.

20. The method in accordance with claim **10** wherein a plurality of Virtual Provisioning Servers are utilized to service a corresponding plurality of multiple administrative areas.

* * * * *

US006515964B1

(12) **United States Patent**
Cheung et al.

(10) **Patent No.: US 6,515,964 B1**
(45) **Date of Patent: Feb. 4, 2003**

(54) **METHOD AND APPARATUS FOR DYNAMICALLY CONTROLLING THE ADMISSION OF CALLS TO A NETWORK**

(75) Inventors: **Hay Yeung Cheung**, Holmdel, NJ (US); **Louise E. Hosseini-Nasab**, Holmdel, NJ (US); **Daniel J. Yaniro, Jr.**, Middletown, NJ (US)

(73) Assignee: **AT&T Corp.**, New York, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/303,305**

(22) Filed: **Apr. 30, 1999**

Related U.S. Application Data

(60) Provisional application No. 60/114,150, filed on Dec. 29, 1998.

(51) **Int. Cl.**[7] **H04L 12/56**
(52) **U.S. Cl.** **370/230**; 370/252; 370/352
(58) **Field of Search** 370/230, 238, 370/352, 356, 252, 253

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,193,151 A	*	3/1993	Jain	370/230
5,400,329 A	*	3/1995	Tokura et al.	370/232
5,732,078 A	*	3/1998	Arango	370/355
5,796,719 A	*	8/1998	Peris et al.	370/230
6,064,653 A	*	5/2000	Farris	370/352
6,192,031 B1	*	2/2001	Reeder et al.	370/230
6,222,824 B1	*	4/2001	Marin et al.	370/230
6,226,266 B1	*	5/2001	Galand et al.	370/235
6,282,192 B1	*	8/2001	Murphy et al.	370/352
6,377,573 B1	*	4/2002	Shaffer et al.	370/356

OTHER PUBLICATIONS

Gordon, Shykeh, H.323: The Multimedia Communications Standard Moves From Consensus to Compliance, CTI Developer, 2(2):108–113.

Blank Michelle, H.323 Gatekeepers: Essential Software for IP Telephony and Multimedia Conferencing, CTI Developer, pp. 94–98, Feb. 1998.

Shenker, et al., RFC 2212 Specification of Guaranteed Quality of Service, pp. 1–20, Sep. 1997.

International Telecommunication Union, H.323, Series H: Audiovisual and Multimedia Systems: Infrastructure of audiovisual services—Systems and terminal equipment for audiovisual services, Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service, 79 pp., Nov. 1996.

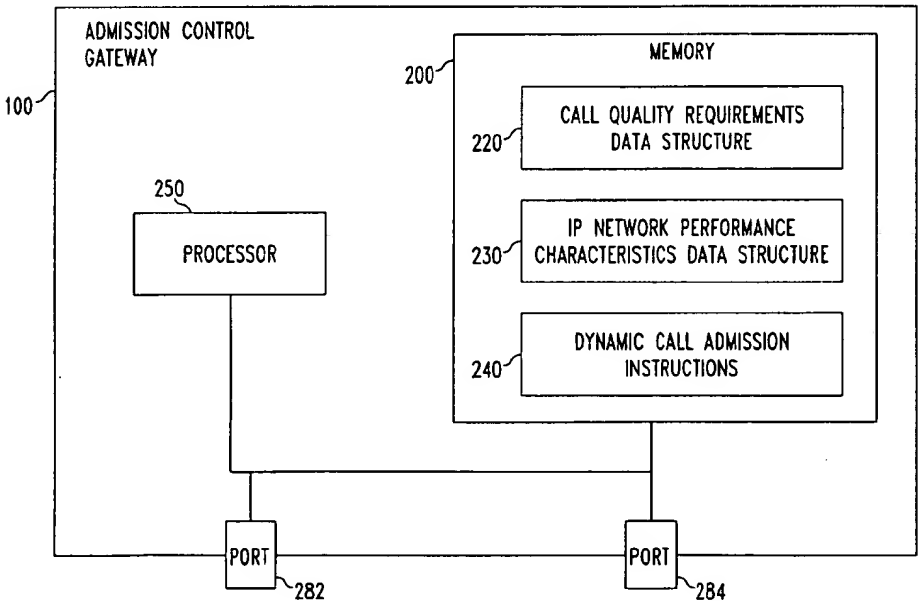
* cited by examiner

Primary Examiner—Chau Nguyen
Assistant Examiner—Keith M. George

(57) **ABSTRACT**

A network call admission control system receives a call and determines a call characteristic requirement and a network characteristic parameter. The call is admitted to the network based in part on whether the call characteristic requirement is satisfied by the network characteristic parameter. As a result, a communications service provider can provide a high quality of service for completed calls or charge a discounted rate for completed calls not meeting a certain quality of service.

3 Claims, 6 Drawing Sheets



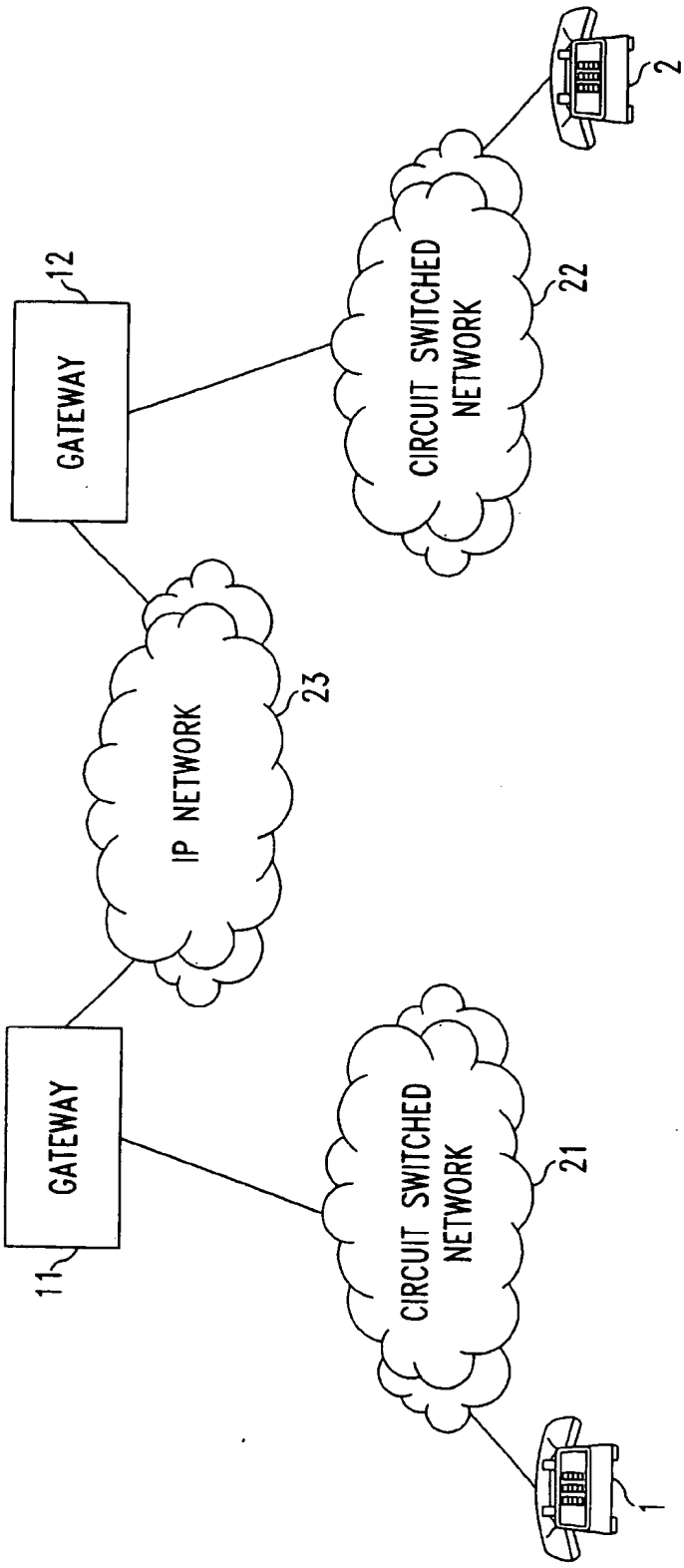


FIG. 1
PRIOR ART

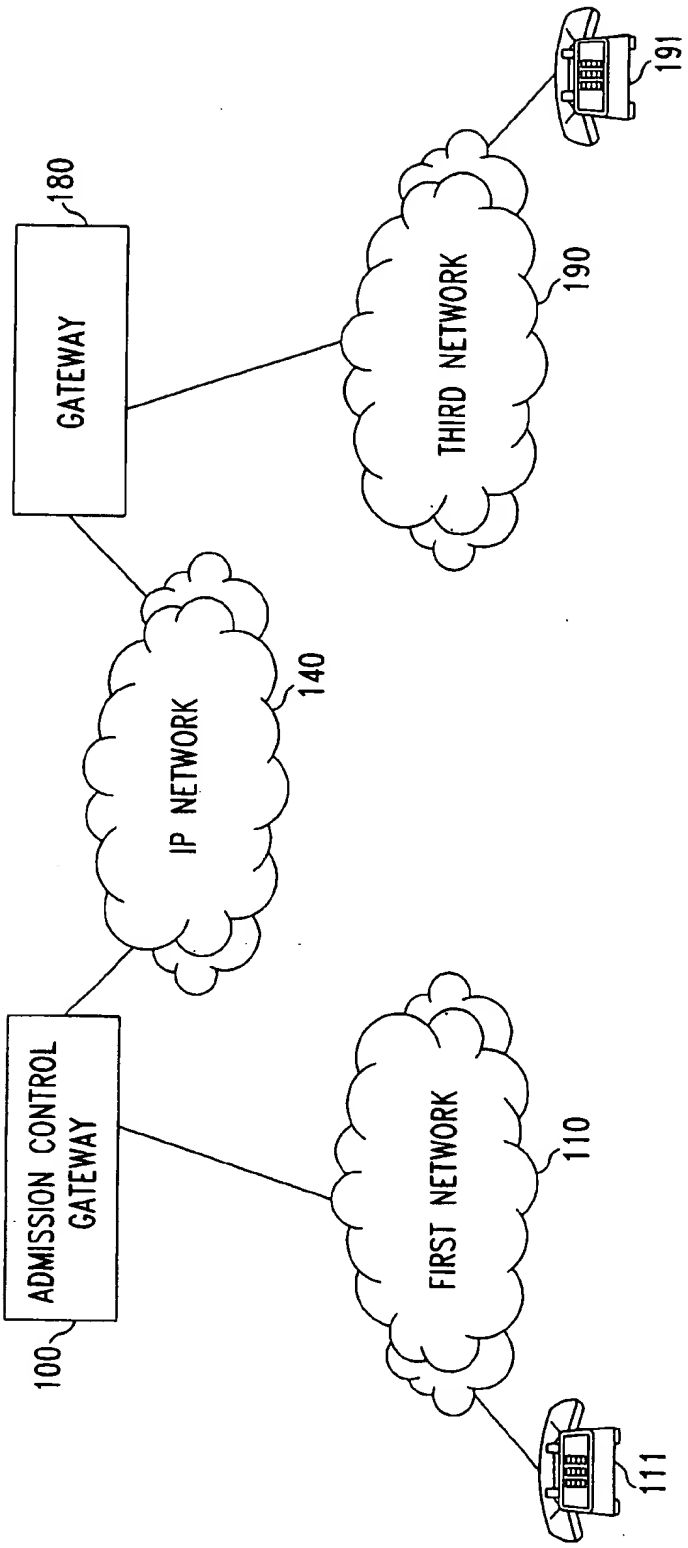
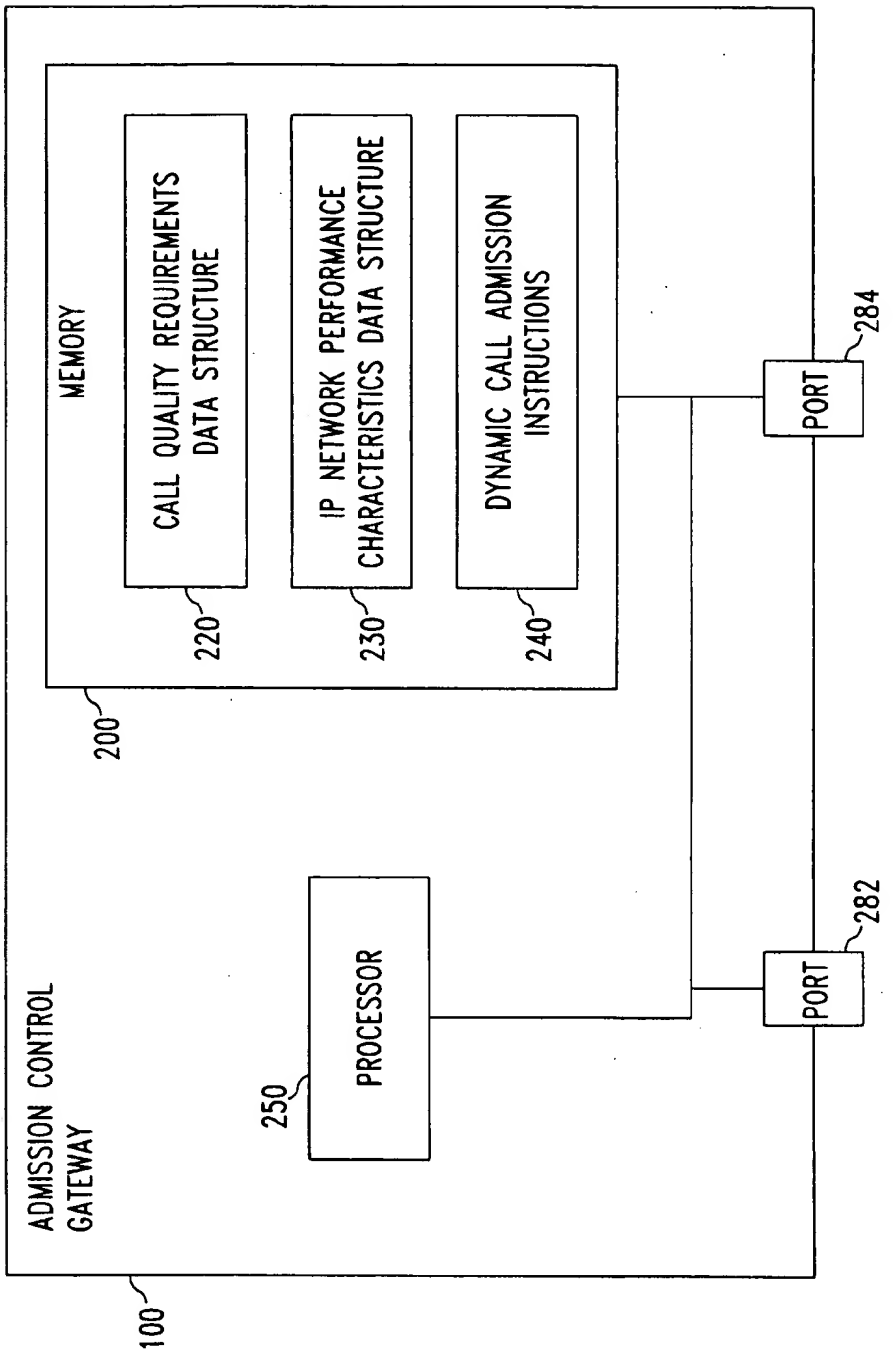

ADMISSION CONTROL GATEWAY
100
GATEWAY
180
IP NETWORK
140
FIRST NETWORK
110
THIRD NETWORK
190
111
191

FIG. 2

FIG. 3


ADMISSION CONTROL GATEWAY
100
MEMORY
200
CALL QUALITY REQUIREMENTS DATA STRUCTURE
220
IP NETWORK PERFORMANCE CHARACTERISTICS DATA STRUCTURE
230
DYNAMIC CALL ADMISSION INSTRUCTIONS
240
PROCESSOR
250
PORT
282
PORT
284

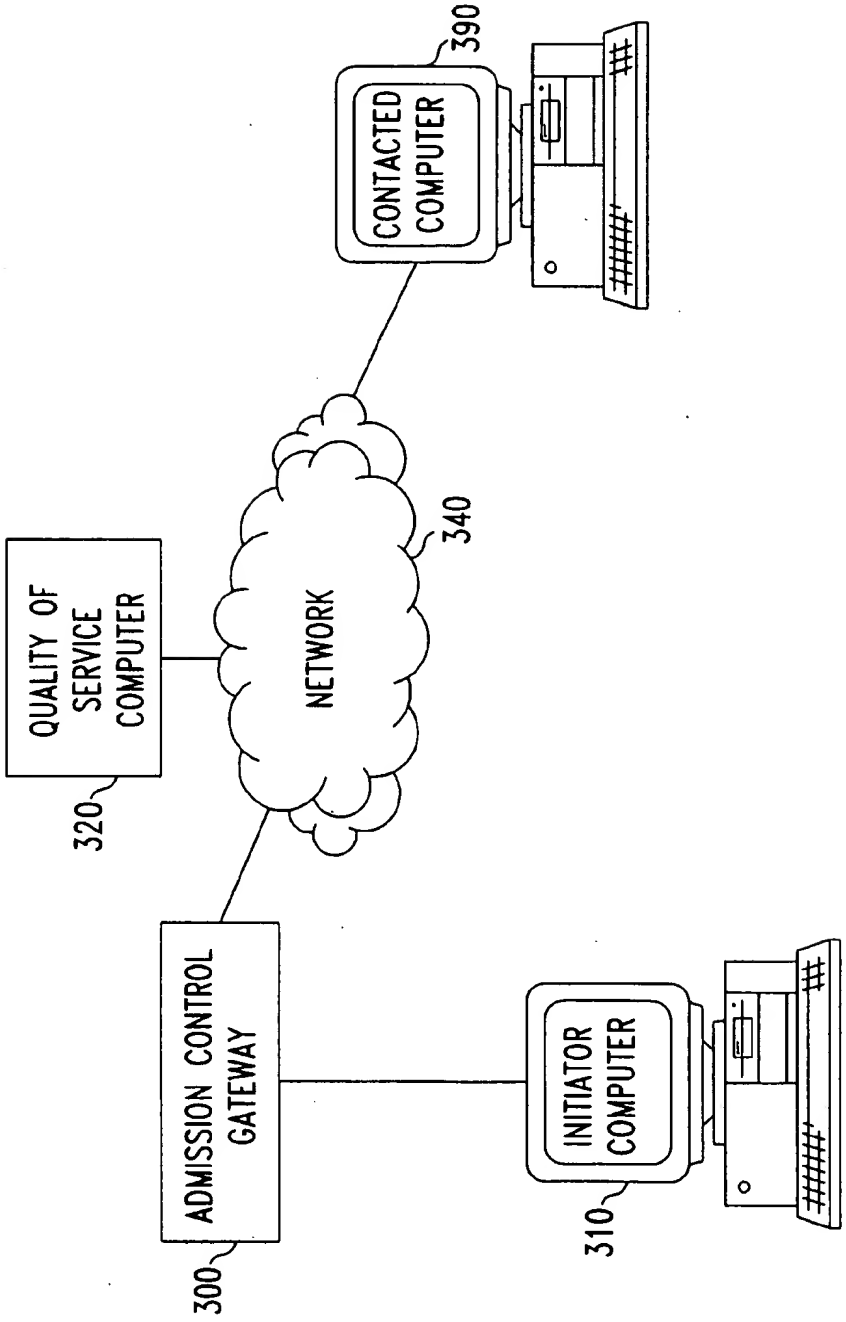
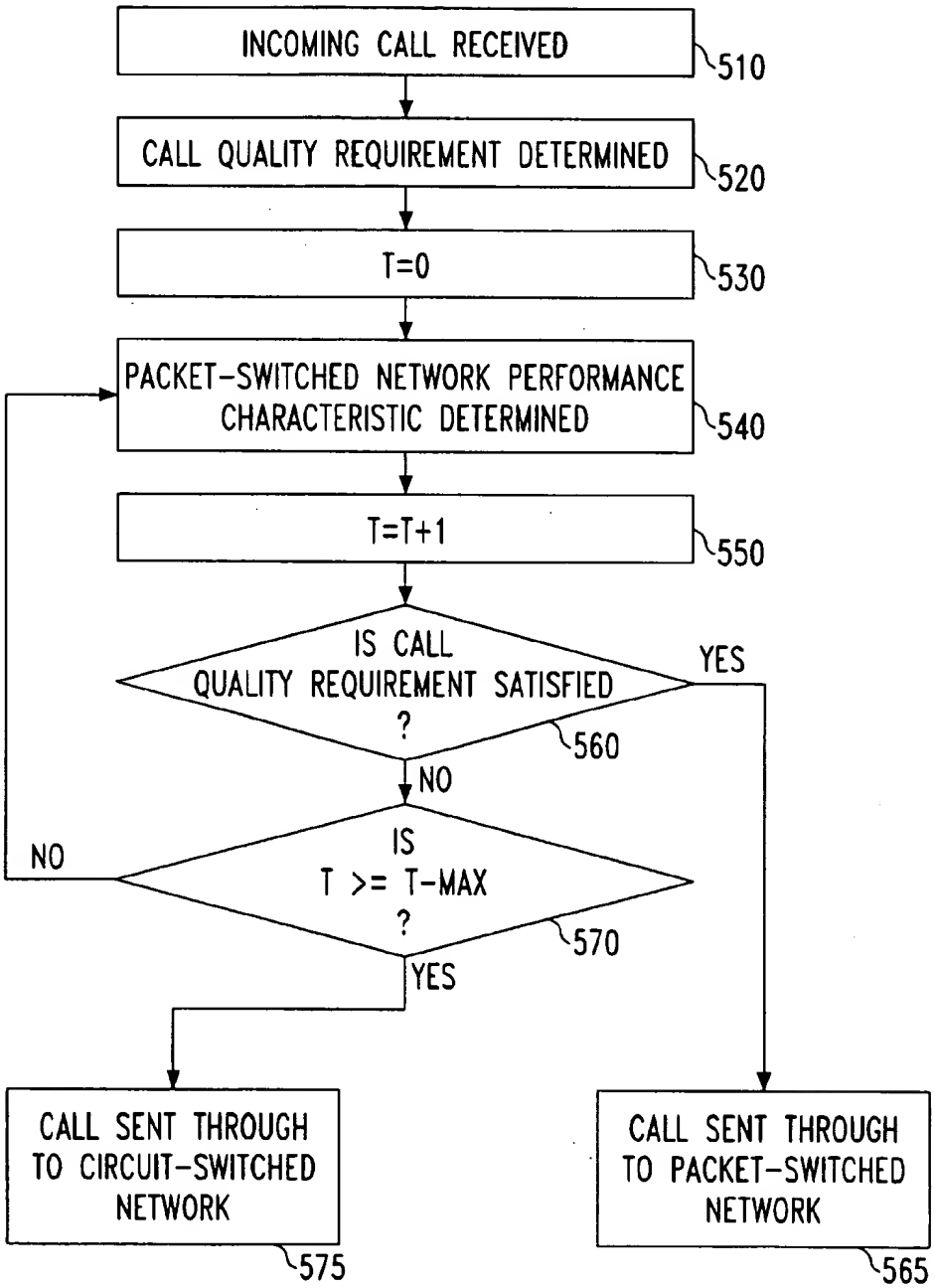

CONTACTED COMPUTER
390
NETWORK
340
QUALITY OF SERVICE COMPUTER
320
ADMISSION CONTROL GATEWAY
300
INITIATOR COMPUTER
310

FIG. 4

FIG. 5


INCOMING CALL RECEIVED
510
CALL QUALITY REQUIREMENT DETERMINED
520
T=0
530
PACKET-SWITCHED NETWORK PERFORMANCE CHARACTERISTIC DETERMINED
540
T=T+1
550
IS CALL QUALITY REQUIREMENT SATISFIED ?
560
YES
NO
IS T >= T-MAX ?
570
NO
YES
CALL SENT THROUGH TO CIRCUIT-SWITCHED NETWORK
575
CALL SENT THROUGH TO PACKET-SWITCHED NETWORK
565

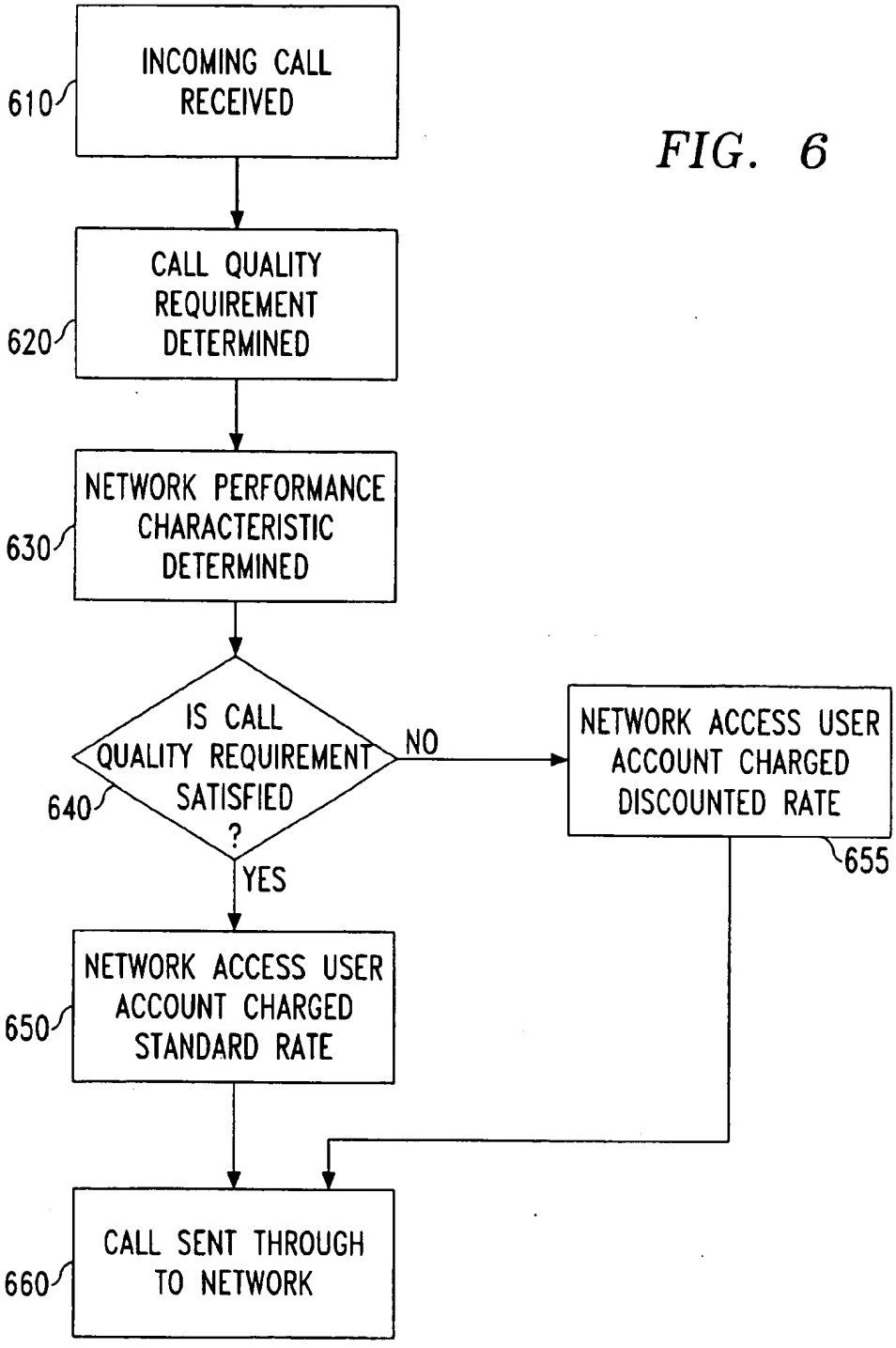

INCOMING CALL RECEIVED
610
CALL QUALITY REQUIREMENT DETERMINED
620
NETWORK PERFORMANCE CHARACTERISTIC DETERMINED
630
IS CALL QUALITY REQUIREMENT SATISFIED ?
640
NO
NETWORK ACCESS USER ACCOUNT CHARGED DISCOUNTED RATE
655
YES
NETWORK ACCESS USER ACCOUNT CHARGED STANDARD RATE
650
CALL SENT THROUGH TO NETWORK
660

FIG. 6

METHOD AND APPARATUS FOR DYNAMICALLY CONTROLLING THE ADMISSION OF CALLS TO A NETWORK

CROSS REFERENCE TO RELATED APPLICATION

The present application claims the benefit of U.S. provisional patent application Ser. No. 60/114,150 entitled "Method and Apparatus for Dynamically Controlling the Admission of Calls to a Network" to Daniel J. Yaniro, Louise E. Brown, and Hay Yeung Cheung and filed on Dec. 29, 1998.

FIELD OF THE INVENTION

The invention relates to network call admission. More particularly, the invention relates to a method and apparatus for dynamically controlling the admission of calls to a network.

BACKGROUND OF THE INVENTION

The present invention relates to a technique for dynamically controlling the admission of traffic to a network based in part on the state of the network.

One known type of network is an Internet Protocol ("IP") network. An IP network implements the protocol specified in RFC 791, Internet Protocol <www.cis.ohio-state.edu/htbin/rfc/rfc791.html, visited November 30, 1998>. One type of traffic carried by known IP networks is voice traffic, called Voice over IP ("VoIP") traffic.

FIG. **1** is an example of a prior art VoIP system. It is known to initiate a voice call from a phone set **1** over a conventional circuit-switched network **21** (such as the public switched telephone network (PSTN)) and route the calling party's voice signals to a first gateway **11** connected to the IP network **23**. The first gateway **11** packetizes the voice signals using the Internet Protocol and transmits the packets as VoIP traffic over the IP network **23** to a second gateway **12** closer to the called party than the first gateway **11**. The packets are converted back into voice signals at the second gateway **12**, and those voice signals are routed via the conventional circuit-switched network **22**, to the called party's phone set **2**.

One of the problems with VoIP services is latency. Latency is the delay between the time a signal is sent and the time it is received. Latency adversely affects the quality of service of real-time communications (e.g., voice communications) and is dependent upon the state of the network over which the communications are carried. For example, a heavily burdened network is likely to have more latency than an underutilized network.

A similar problem arises in the context of users making other types of calls over a packet-switched network, such as the Internet. At present, a user can be connected to the Internet by an Internet Service Provider (ISP) and can make a number of calls over the Internet via HTTP (Hypertext Transfer Protocol) commands (using a Web browser such as Microsoft Internet Explorer or Netscape Navigator), FTP (File Transfer Protocol) commands, TELNET connections, and the like. The user may encounter significant delays in accessing, for example, Web sites. Those delays can be caused by a number of factors, including a Web site's inability to respond to all of the users that concurrently seek information from that Web site. A user also may experience significant delays in accessing a particular Web site, not due to that Web site's inability to meet the demand for that site, but due to poor performance characteristics of one or more networks which couple the user to the Web site, or the internetwork routers.

In known VoIP systems, a gateway will pass traffic into a network whenever the gateway has an incoming port that is available to do so. Thus, certain networks must disadvantageously be over-engineered to be able to carry a peak load equal to the traffic that flows when all of the ports of all of the gateways connected to the network are in use. If the traffic sent through the network approaches or exceeds the network's capacity, then the network disadvantageously drops packets (i.e., experiences packet loss) and/or introduces unacceptable delays into communications. In known networks, it is difficult or impossible to guarantee a high quality of service when the network is operating near or at its capacity.

The International Telecommunications Union ("ITU") has established the H.323 standard, which encompasses audio, video and data communications across packet-switched networks, such as the Internet. The H.323 standard was principally developed and established to allow multimedia products and applications from multiple vendors to interoperate. H.323 systems may include a gatekeeper, which can provide bandwidth management. For example, the gatekeeper can reject calls from a terminal if it determines that sufficient bandwidth is not available. H.323 bandwidth management also operates during an active call if a terminal requests additional bandwidth, and the gatekeeper may grant or deny the request for additional bandwidth. Likewise, there are other Internet protocols that provide for establishing or rejecting calls based on bandwidth requirements (e.g., RFC 2211, Specification of the Controlled-Load Network Element Service, <www.cis.ohio-state.edu/htbin/rfc/rfc2211.html, visited Jan. 11, 1999>; RFC 2210, The Use of RSVP with IETF Integrated Services, <www.cis.ohio-state.edu/htbin/rfc/rfc2211.html, visited Jan. 11, 1998>). These bandwidth management protocols do not provide for admitting or rejecting calls based on delay characteristics of the network.

SUMMARY OF THE INVENTION

The present invention provides a system for regulating the call traffic into a packet-switched network based in part upon delay characteristics of the network. In an embodiment of the present invention, a call delay characteristic requirement for a call is determined, a delay characteristic parameter of the packet-switched networks is determined, and a call action based at least partly upon the determined delay characteristic requirement and the determined delay characteristic parameter is performed.

In one embodiment of the invention, the network is an Internet Protocol (IP) network carrying Voice over IP (VOIP) traffic. A voice call made in connection with a VoIP service is not admitted to the IP network and is held if one or more current delay characteristic parameters of the IP network do not satisfy one or more prescribed delay characteristic requirements. Delay characteristic parameters can be periodically updated, and when the current value of one or more delay characteristic parameters satisfy one or more prescribed delay requirements, the VoIP call is admitted to the IP network.

Another embodiment of the present invention dynamically controls the admission of other traffic to an IP network, including multimedia communications, HTTP commands, FTP commands, TELNET connections, and the like. This embodiment allows such data calls to be admitted to the IP network when the IP network satisfies the delay requirements.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG; 1 is an illustration of a prior art Voice over IP system.

FIG. 2 shows a system in accordance with an embodiment of the present invention.

FIG. 3 shows an apparatus in accordance with an embodiment of the present invention.

FIG. 4 shows a system with a quality of service computer in accordance with an embodiment of the present invention.

FIG. 5 is a flowchart illustrating a method in accordance with an embodiment of the present invention whereby a voice call is rerouted over another network when the call quality requirements are not satisfied within a set maximum time.

FIG. 6 is a flowchart illustrating a method in accordance with an embodiment of the present invention that can be utilized as part of a service that markets a certain level of network performance to network users placing calls over the network.

DETAILED DESCRIPTION

A method and apparatus for dynamically controlling the admission of calls to a packet-switched network is described. In the following description, for purposes of explanation, numerous specific details are set forth to provide a thorough understanding of the present invention. It will be obvious, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well known structures and devices are shown in block diagram form and process steps are shown in flowcharts to describe the present invention. Furthermore, it is readily apparent to one skilled in the art that the specific sequences in which steps are presented and performed are illustrative and it is contemplated that the sequences can be varied and still remain within the spirit and scope of the present invention.

A packet-switched network is meant to encompass any network that routes information in the form of discrete packets. For example, a packet may have a header and a payload. The header can include routing and contextual information, which can include the sender's address, the destination address, and a packet number that indicates the packet's place in a series of packets that together comprise a set of information such as a message or a file. Packets can be routed in a connectionless or connection-oriented fashion. In a connectionless protocol such as the Internet Protocol, the path taken by a packet is determined at each router based upon the packet's destination address and conditions in the network. In a connection-oriented protocol, such as the Asynchronous Transfer Mode ("ATM") protocol, a packet is routed along a predetermined path (a circuit) through the network based upon a circuit number that is assigned to the packet.

As used herein, the term "packet-switched network" is meant to encompass connectionless packet-switched networks, connection-oriented networks, and any network that employs a combination of connectionless and connection-oriented protocols to route packets.

As used to describe the present invention, a call is meant to encompass any communication that is carried by a network between entities that are coupled by that network. An entity is anything adapted to utilize a network to communicate with any other entity. Examples of an entity include a telephone, a computer, a facsimile machine, etc. For example, a voice call includes a communication that is carried by a network between a calling party and a called party. As used herein, admitting a call to a network means permitting the network to carry the call.

In an embodiment of the present invention, a calling party initiates a voice call over a conventional circuit-switched network. The incoming voice call is routed over a circuit-switched network from the calling party to a gateway. The gateway decides whether to admit the call to the packet-switched network based in part on the state of the packet-switched network. For example, if the network is already overburdened, the gateway will not admit the call to the network. If, on the other hand, the network can carry the call with an appropriate quality of service, the gateway will admit the call to the network.

The state of a packet-switched network can be indicated by a number of performance parameters, including total delay, mean and standard deviation for such delay, packet loss, error rate, etc. These network characteristic parameters can be determined by methods well known in the art. For example, total delay is the time interval from when one party utters a sound to when the other party hears that sound. It can be determined by methods well known in the art including the timed transmission of audible tones. Packet loss is the percentage of packets transmitted but not received, and can be measured by sending a known set of packets and determining how many are received.

Call quality requirements for the various performance parameters of the packet-switched network (e.g., total delay, mean and standard deviation for such delay, and packet loss) can be established to enable a higher quality of service for certain calls. For example, one delay characteristic requirement is a typical delay requirement, which can require that the network's typical delay parameter be below a certain maximum value before the gateway admits the call to the packet-switched network. As used to describe the call delay characteristic requirements and network delay characteristic parameters, the meaning of the term "typical delay" encompasses an average delay, a mean delay, a median delay, an arithmetic mean delay, a weighted average delay, and other derived delay values that represent a practicable expected delay value. For example, one type of a typical delay requirement is a maximum mean delay requirement, and one type of a typical delay parameter is a mean delay parameter.

Another call delay characteristic requirement is a delay variation requirement, which can require that the network's delay variation parameter be below a certain maximum value before the gateway admits the call to the packet-switched network. As used herein, the meaning of the term "delay variation" encompasses a delay standard deviation, other order moments of the delay distribution, a delay variance, a delay coefficient of skewness, a delay kurtosis, a delay covariance, a delay range, a delay standard error, a delay maximum, a delay minimum, and other derived delay values that represent a practicable delay variation value. For example, one type of a delay variation requirement is a maximum delay standard deviation requirement, and one type of a delay variation parameter is a delay standard deviation parameter.

A call quality requirement (e.g., delay characteristic requirement) may be particular to certain types of calls, call services, the calling party, the called party, and other call differentiations known to one skilled in the art.

Each incoming voice call to the packet-switched network can be held if the current values of the performance parameters are outside the prescribed call quality requirements. As each incoming call is held, actual values of the performance parameters are updated. Various call actions can be taken

while the voice call is held (e.g., sending a wait message to the calling party, sending the calling party a ringing message), and various call actions can be taken if the voice call cannot be admitted to the packet-switched network (e.g., holding the voice call, sending the calling party a busy signal, providing the calling party the option of having the system call him or her back when the VoIP call can be admitted to the network, or rerouting the voice call over another network, such as a conventional circuit-switched network). Voice calls are admitted to the packet-switched network when the current values of the performance parameters are within the prescribed call quality requirements. Once a voice call is admitted to the packet-switched network, all packets associated with the call can be permitted to proceed back and forth through the network as the calling party and called party converse.

While one embodiment of the present invention concerns VoIP services, other embodiments of the present invention concern the admission of any type of call to a packet-switched network. Other types of calls encompassed by the present invention include multimedia communications (e.g., video phone calls), HTTP commands, FTP commands, TELNET connections, and other calls that concern the transmission of data across a packet-switched network. As used to describe the present invention, multimedia communications include audio, video, graphics, animation, facsimile, text communication, and any combination thereof.

FIG. **2** shows a VoIP system which operates in accordance with an embodiment of the present invention. Referring to FIG. **2**, the system includes an admission control gateway **100** that is coupled to a first network **110**, such as a PSTN or a private branch exchange (PBX). Connected to the first network **110** is a telephony station **111**. Examples of such a telephony station include a conventional telephone, a wireless telephone station, a personal computer system with a microphone and headphones, a video conferencing system, a facsimile machine containing a phone handset, etc. The admission control gateway **100** is also coupled to an IP network **140**, which is also connected to a second gateway **180**. A third network **190** is connected to the second gateway **180** and to a telephony station **191**. Networks **110** and **190** may be separate telephone networks or different parts of the same telephone network.

Admission control gateway **100** performs functions that are well known in the art, including receiving from the first network **110** voice signals from a voice call initiated at telephony station **111**, packetizing the voice signals using the Internet Protocol, and transmitting the packets over the IP network **140**. Gateway **180** also performs functions that are well known in the art, including receiving from the IP network **140** packets containing packetized voice signals, converting those packets into voice signals, and routing the voice signals over the third network **190** to the called telephony station **191**. Moreover, gateways **100** and **180** can receive and route other data calls, such as those associated with multimedia communications, HTTP commands, FTP commands, TELNET connections, etc. As used to describe the present invention, the admission control gateway **100** receives one type of a data call when it receives data from the first network **110** to be transmitted over the IP network **140**. A gateway to a packet-switched network also receives a data call when it receives packets of data from another network (e.g., a conventional circuit-switched network, an ATM network, an IP network, etc.) to be transmitted over the packet-switched network. Each gateway can also accumulate data parameters about the network and the current traffic, including network performance parameters, e.g., by polling every other gateway in the network and/or receiving data from network components, such as routers (not shown). Hence, each gateway is able to keep or access up-to-date network data parameters.

The admission control gateway **100** can place dynamic controls on the calls that are to be admitted to the IP network **140** at any given time. Quality of service can be monitored and access controlled to allow calls into the IP network when acceptable service is assured. Call quality for Voice over IP calls can thereby be maintained at an acceptable level. Customer complaints regarding poor quality calls can be reduced. Overengineering of facilities and other resources can be minimized, saving capital and expense.

Referring to FIG. **3**, the admission control gateway **100** includes a processor **250** and a memory **200**. The processor **250** in one embodiment is a general purpose microprocessor, such as the Pentium II processor manufactured by the Intel Corporation of Santa Clara, Calif. In another embodiment, the processor **250** is an Application Specific Integrated Circuit (ASIC), which has been designed to perform in hardware and firmware at least part of the method in accordance with an embodiment of the present invention. Memory **200** is any device adapted to store digital information, such as Random Access Memory (RAM), flash memory, a hard disk, an optical digital storage device, any combination thereof, etc. As shown in FIG. **3**, memory **200** is coupled to processor **250**, a port **282** adapted to be coupled to a sender of a call (e.g., a circuit-switched network), and a port **284** adapted to be coupled to a packet-switched network. The term "coupled" means connected directly or indirectly. Thus, A is "coupled" to C if A is directly connected to C, and A is "coupled" to C if A is connected directly to B, and B is directly connected to C.

In accordance with one embodiment of the present invention, dynamic network call admission instructions are stored on a medium and distributed as software. The medium is any device adapted to store digital information, and corresponds to memory **200**. For example, a medium is a portable magnetic disk, such as a floppy disk; or a Zip disk, manufactured by the Iomega Corporation of Roy, Utah; or a Compact Disk Read Only Memory (CD-ROM) as is known in the art for distributing software. The medium is distributed to a user that has a processor suitable for executing the dynamic network call admission instructions, e.g., to a user with a gateway having a processor, memory, a port adapted to be coupled to a circuit-switched network, and a port adapted to be coupled to a packet-switched network.

Exemplary data structures and instructions adapted to be executed by a processor stored in the memory **200** include the call quality requirements data structure **220**, the packet-switched network performance parameters data structure **230**, and the dynamic call admission instructions **240**.

The call quality requirements data structure **220** can contain the call quality requirements for all calls, certain types of calls, and/or each individual call. For example, a call may have a maximum delay bound, d, for call connection. The delay, d, has mean μ and standard deviation σ. A maximum bound for packet loss, defined by ρ, can also be included in the call quality requirements data structure **220**. Other maximum bounds may be established for error rates and other network performance parameters concerning the IP network, current network traffic and projected network traffic. The call quality requirements data (e.g., the maximum delay bound d) can be predetermined for all calls received by the gateway and stored in the call quality requirements data structure **220**. Alternatively, the call qual-

ity requirements data may be stored in a lookup table that specifies certain call quality requirements for certain types of calls, specific calling parties, specific called parties, etc. For example, the call quality requirements data structure **220** can contain a lookup table indexed according Automatic Number Information (ANI) of the call. The gateway utilizes the calling party's ANI to determine the call quality requirements data for that call from the lookup table. Furthermore, the call quality requirements data can vary for each call by including the call quality requirements data in the call signal itself. The gateway then extracts or reads the call quality requirements data from each call and then stores that data in the call quality requirement data structure **220**.

The packet-switched network performance parameters data structure **230** includes current and/or projected performance parameters of the network. Examples of network performance parameters data include call delay, packet loss, error rate, etc. The delays associated with a call include d_1, d_2, and d_3: d, is the time taken by the incoming gateway to packetize the voice signals; d_2 is the time taken by the outgoing gateway to reassemble the packets into voice signals; and d_3 is the time taken to relay the packets through the IP network and has a standard deviation σ_3. In one embodiment, the incoming and outgoing gateways are the gateways closest to the calling party and the called party, respectively. The delays d_1 and d_2 are functions of $\lambda_i^{(j)}$ and $\lambda_o^{(j)}$, the rate of incoming and outgoing calls at gateway j, respectively. The packet loss p and the delay d_3 are both functions of the IP network and traffic. Although d_1 and d_2 have means μ_1 and μ_2 and standard deviations σ_1 and σ_2 respectively, the standard deviations σ_1 and σ_2 are assumed to be small so that the actual delay, $d_{actual}=(d_1+d_2+d_3)$ has mean $\mu=(\mu_1+\mu_2+\mu_3)$ and standard deviation σ_3.

The network performance parameters data can be accumulated by each gateway such that each gateway keeps up-to-date data. Alternatively, the gateway ascertains the network performance parameters data by accessing quality of service computer **320** of FIG. **4** that determines the appropriate data for each gateway.

Memory **200** stores the dynamic call admission instructions **240** adapted for execution by processor **250**. The term "adapted for execution" is meant to encompass any instructions that are ready for execution in their present form (e.g., machine code) by processor **250**, or require further manipulation (e.g., compilation, decryption, or provided with an access code, etc.) to be ready for execution by processor **250**. The dynamic call admission instructions **240** can determine and indicate when a call has been received by the admission control gateway **100**. The call can then be placed into a queue or otherwise controlled. In some applications of the present invention, the dynamic call admission instructions **240** track how much time has passed since a call was received by the admission control gateway **100** so that the appropriate call action can be taken. In addition, the dynamic call admission instructions **240** can ascertain and make available information about the call such as calling party's ANI, the called party's phone number or IP address, and the type of call requested (e.g., voice call, data call, etc.).

The dynamic call admission instructions **240** also can determine for each call the appropriate call action based upon a call quality requirement and a network performance parameter. For example, a call has maximum bounds for delay and packet loss, which are $d(\mu_{maximum}, \sigma_{maximum})$ and $\rho_{maximum}$, respectively. One call action may be to hold a call in a queue if it is determined that any of the current measured parameters μ, σ_3, and ρ exceed the maximum boundaries. Furthermore, as the call is held in queue, μ, σ_3, and ρ are updated every t_k seconds. The call is admitted to the network when the updated parameters are less than the respective maxima.

In particular, in one embodiment of the present invention, a VoIP call arrives at the gateway at time t_k. The VoIP call is admitted to the IP network based on the following steps:

1. The call characteristic requirements data $\mu_{maximum}$, $\sigma_{maximum}$, and $\rho_{maximum}$, are determined;
2. The network characteristic parameters data $\mu(t_k)$, $\sigma(t_k)$, and $\rho(t_k)$ are determined;
3. If $\mu(t_k) \leqq \mu_{maximum}$, $\sigma(t_k) \leqq \sigma_{maximum}$, and $\rho(t_k) \leqq \rho_{maximum}$, the call is admitted to the IP network;
4. If any one of $\mu(t_k)$, $\sigma(t_k)$, or $\rho(t_k)$ exceeds its respective maximum bound, then the call is held in a queue; and
5. At t_{k+1}, set k+1=k and go to step 1.

In this particular embodiment, the VoIP call is held in the queue until the call is admitted to the IP network. Alternative call actions can provide that the call is held in the queue for a specified amount of time; and if the call has not been admitted to the IP network within that time, then the call is routed over a conventional circuit-switched network to the called party. Another call action can provide for a call back to the calling party when the call can be admitted to the IP network. Instead of holding the call in the queue, the call action can also be to send the calling party a distinctive busy signal or message that indicates that the IP network cannot handle the call at the present moment.

FIG. **4** is an example of another embodiment of the present invention that admits calls from an initiator computer **310** to a contacted computer **390** when the parameters of network **340** satisfy certain requirements. In this embodiment, the calls from the initiator computer **310** can be voice calls and/or data calls such as multimedia communications, HTTP commands, FTP commands, TELNET commands, etc. The admission control gateway **300** receives the call from the initiator computer **310**, determines the call quality requirements, determines the network performance parameters from information provided by a quality of service computer **320**, and takes a call action based on the determined call quality requirements and network performance parameters. The quality of service computer **320** is able to keep up-to-date data parameters about the network, the current traffic, and/or projected traffic by methods well known in the art, including polling every other gateway in the network, receiving data from network components such as routers (not shown), and/or accessing data concerning historical and cyclical traffic patterns (e.g., peak voice call traffic occurs between certain hours weekdays, peak residential data calls occur between certain hours each day, etc.).

FIG. **5** illustrates exemplary steps whereby an embodiment of the present invention reroutes a voice call over another network, such as a conventional circuit-switched network, when the call quality requirements are not satisfied within a set maximum time measured after the voice call is received by the admission control gateway. After an incoming call is received (step **510**), call quality requirements are determined (e.g., a typical delay requirement, a delay variation requirement) (step **520**) and time variable T is set to equal 0 (step **530**). The packet-switched network performance parameters are determined (e.g., a typical delay parameter, a delay variation parameter) (step **540**), and time variable T is incremented (step **550**). The determined network performance parameters are compared to the determined call quality requirements to ascertain whether the call quality requirements are satisfied (step **560**). If the network performance parameters satisfy the call quality

requirements, the call is admitted to the packet-switched network (step **565**). On the other hand, if the call quality requirements are not satisfied, the time variable T is compared to a certain maximum value (step **570**). If T equals or exceeds the maximum value, then the call is routed to a conventional circuit-switched network (step **575**). If T does not equal or exceed the maximum value, then the packet-switched network performance parameters are determined again (step **540**), the time variable T is again incremented (step **550**), etc. Thus, the call is admitted to the packet-switched network if the call quality requirements are satisfied within a certain amount of time, or the call is routed through to a conventional circuit-switched network if the call quality requirements are not satisfied within a certain amount of time.

FIG. **6** illustrates exemplary steps whereby an embodiment of the present invention can be used in conjunction with a service that markets a certain level of network performance to network users placing calls over the network. An account for each user (e.g., a network access user account) may be charged a reduced rate if the network performance for a call is below the marketed level of performance. In particular, and referring to FIG. **6**, after an incoming call is received (step **610**), call quality requirements are determined (e.g., a typical delay requirement, a delay variation requirement) (step **620**); network performance parameters are determined (e.g., a typical delay parameter, a delay variation parameter) (step **630**); and the determined network performance parameters are compared to the determined call quality requirements to ascertain whether the call quality requirements are satisfied (step **640**). If the network performance parameters do not satisfy the call quality requirements, the network access user account is charged a discounted rate (step **655**) and the call is admitted to the network (step **660**). On the other hand, if the call quality requirements are satisfied, the network access user account is charged the standard rate (step **650**) and the call is admitted to the network (step **660**).

In another embodiment of the invention, after a first call action is taken to admit a call to the packet-switched network, a second call action can be performed when the performance parameters of the network no longer satisfy the call quality requirements. For example, periodically during the call (e.g., at specific time intervals) the determined call quality requirements are compared to updated, determined performance parameters to ascertain whether the call quality requirements are still satisfied. When the call quality requirements are no longer satisfied, a second call action can reroute the call over another network. Another second call action may be to charge a discounted rate for the call if the network performance parameters do not satisfy the call quality requirements at a point during the call. Alternatively, the second call action may be to charge a discounted rate for the period of the call during which the network performance parameters do not satisfy the call quality requirements.

Thus, the present invention provides a method and apparatus whereby the admission of calls into a packet-switched network can be dynamically controlled. Performance of the network can be monitored and the admission of a call to the network is controlled to allow the call into the network when a certain level of call quality is met. Call quality can thereby be maintained at an acceptable level.

The invention has been described in conjunction with the preferred embodiment. It is evident that numerous alternatives, modifications, and uses will be apparent to those skilled in the art in light of the foregoing description.

What is claimed is:

1. A method for regulating the admission of a call to a packet-switched network, comprising:

determining a delay characteristic requirement of the call, said requirement including a typical delay requirement and a delay variation requirement;

determining a delay characteristic parameter of the packet-switched network, said parameter including a typical delay parameter and a delay variation parameter; and

performing a call action based at least partly upon whether the determined delay characteristic parameter satisfies the determined delay characteristic requirement, wherein the performing of the call action includes:

admitting the call to the packet-switched network if:

i. the determined typical delay parameter does not satisfy the determined typical delay requirement; or

ii. the determined delay variation parameter does not satisfy the determined delay variation requirement; and

charging a reduced rate for the call.

2. A method for regulating the admission of a call to a packet-switched network, comprising:

determining a delay characteristic requirement of the call, said requirement including a typical delay requirement and a delay variation requirement;

determining a delay characteristic parameter of the packet-switched network, said parameter including a typical delay parameter and a delay variation parameter;

performing a call action based at least partly upon whether the determined delay characteristic parameter satisfies the determined delay characteristic requirement, wherein the performing of the call action includes:

admitting the call to the packet-switched network if:

the determined typical delay parameter satisfies the determined typical delay requirement; and

the determined delay variation parameter satisfies the determined delay variation requirement;

updating the determined typical delay parameter and the determined delay variation parameter after the call is admitted to the network; and

performing a second call action based at least partly upon whether the updated determined typical delay parameter satisfies the determined typical delay requirement and whether the updated determined delay variation parameter satisfies the determined delay variation requirement wherein the performing of a second call action includes charging a reduced rate for the call if:

i. the updated determined typical delay parameter does not satisfy the determined typical delay requirement; or

ii. the updated determined delay variation parameter does not satisfy the determined delay variation requirement.

3. An apparatus for controlling the admission of a call into a packet-switched network, comprising:

a processor; and

a memory, coupled to said processor, storing a plurality of instructions adapted for execution by said processor to:

determine a delay characteristic requirement of the call, said requirement including a typical delay requirement and a delay variation requirement;

determine a delay characteristic parameter of the packet-switched network, said parameter including a typical delay parameter and a delay variation parameter;

perform a call action based at least partly upon whether the delay characteristic parameter satisfies the delay characteristic requirement, wherein said instructions for performing the call action includes instructions to admit the call to the packet-switched network if:

the determined typical delay parameter satisfies the determined typical delay requirement; and

the determined delay variation parameter satisfies the determined delay variation requirement;

update the determined typical delay parameter and the determined delay variation parameter after the call is admitted to the network; and

perform a second call action based at least partly upon whether the updated determined typical delay parameter satisfies the determined typical delay requirement and whether the updated determined delay variation parameter satisfies the determined delay variation requirement wherein said instructions for performing the second call action includes instructions to charge a reduced rate for the call if:

i. the updated determined typical delay parameter does not satisfy the determined typical delay requirement; or

ii. the updated determined delay variation parameter does not satisfy the determined delay variation requirement.

* * * * *

US006377573B1

(12) **United States Patent**
Shaffer et al.

(10) **Patent No.: US 6,377,573 B1**
(45) **Date of Patent: Apr. 23, 2002**

(54) **METHOD AND APPARATUS FOR PROVIDING A MINIMUM ACCEPTABLE QUALITY OF SERVICE FOR A VOICE CONVERSATION OVER A DATA NETWORK**

(75) Inventors: **Shmuel Shaffer**, Palo Alto; **William Joseph Beyda**, Cupertino, both of CA (US)

(73) Assignee: **Siemens Information and Communication Networks, Inc.**, Boca Raton, FL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/097,846**

(22) Filed: **Jun. 15, 1998**

(51) **Int. Cl.**[7] **H04L 12/66**; G06F 15/16; G10L 21/04; H04B 1/38

(52) **U.S. Cl.** **370/356**; 370/521; 704/500; 709/247; 455/553

(58) **Field of Search** 370/351–352, 370/356, 353–354, 521, 468, 493–495; 379/88.1, 93.08; 704/500, 502, 503; 714/820–822, 1–2, 6, 11, 48; 455/66, 74, 72, 101, 553; 375/299, 347; 709/247

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,546,395 A		8/1996	Sharma et al.	370/468
5,550,893 A		8/1996	Heidari	455/553
5,742,773 A	*	4/1998	Blomfield-Brown et al.	704/500
6,006,179 A	*	12/1999	Wu et al.	704/205
6,151,636 A	*	11/2000	Schuster et al.	709/247
6,178,405 B1	*	1/2001	Ouyang et al.	704/500

FOREIGN PATENT DOCUMENTS

WO	WO 97/27692	* 7/1997	H04L/12/56

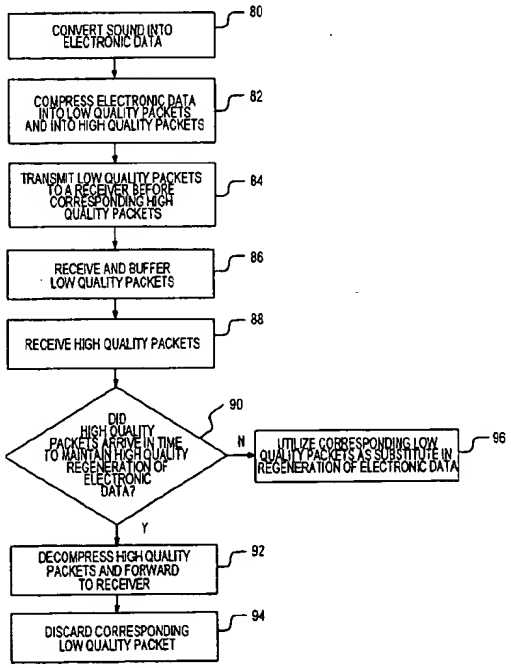
* cited by examiner

Primary Examiner—Wellington Chin
Assistant Examiner—Maikhanh Tran

(57) **ABSTRACT**

A method and apparatus for transmitting delay-sensitive data over a packet-based network involve converting the delay-sensitive data into two versions for transfer through the network with one version of the data being used to supplement the other version of the data in the event that packets are delayed or lost. In a preferred embodiment, real time voice conversation data is compressed using two different compression algorithms, where one version is more highly compressed and of a lower quality than the other. The highly compressed data is sent before corresponding packets from the less compressed data and the highly compressed packets are buffered at the receiving device to be utilized to regenerate any data segments from the less compressed packets that are lost or delayed during transmission. Sending dual versions of the same data allows the lower quality voice segment to be used as a backup in places where the high quality voice segments are lost.

20 Claims, 3 Drawing Sheets



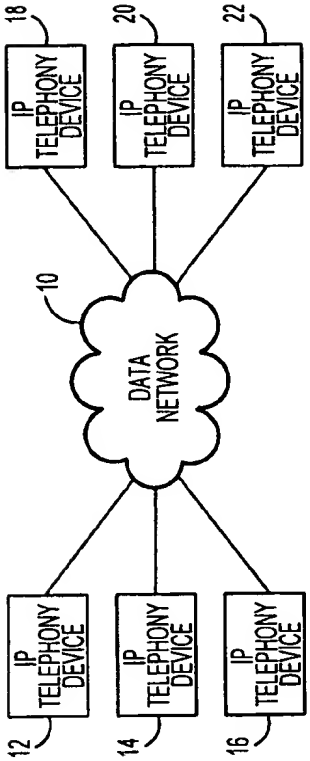

IP TELEPHONY DEVICE
18
IP TELEPHONY DEVICE
20
IP TELEPHONY DEVICE
22
10
DATA NETWORK
IP TELEPHONY DEVICE
12
IP TELEPHONY DEVICE
14
IP TELEPHONY DEVICE
16

FIG. 1

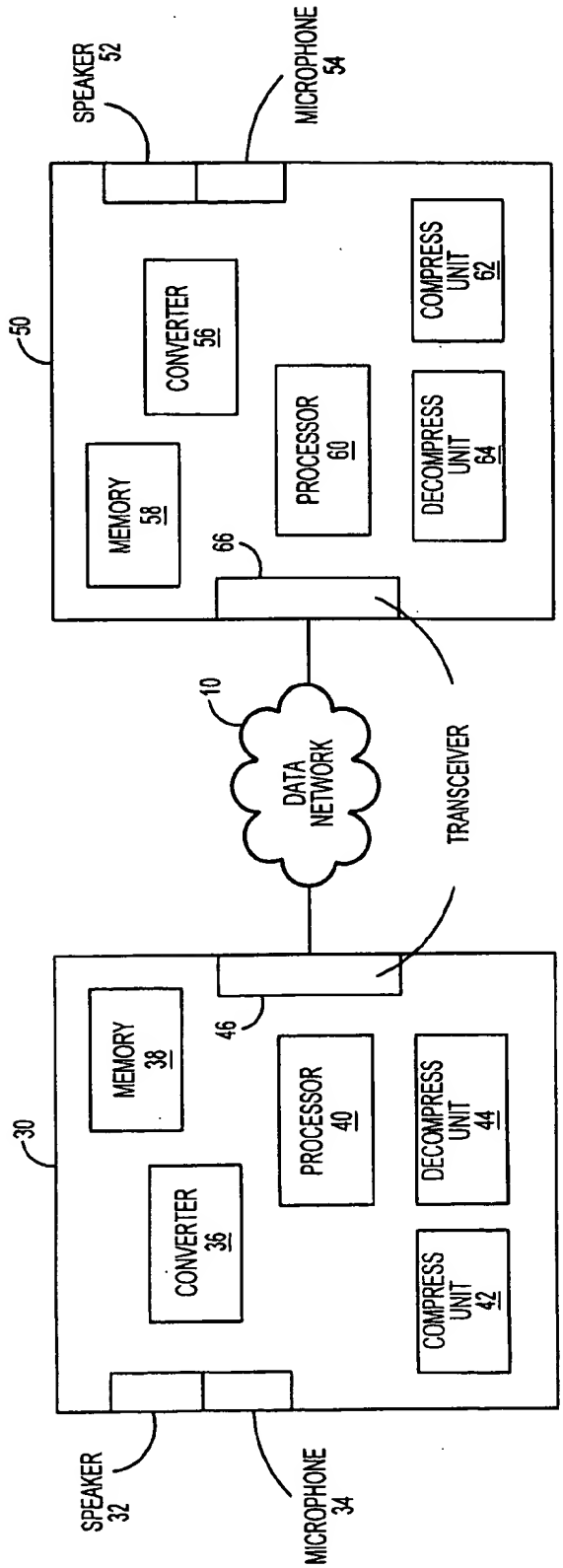

SPEAKER 52
MICROPHONE 54
50
CONVERTER 56
COMPRESS UNIT 62
MEMORY 58
PROCESSOR 60
DECOMPRESS UNIT 64
66
10
DATA NETWORK
TRANSCEIVER
46
30
MEMORY 38
PROCESSOR 40
DECOMPRESS UNIT 44
CONVERTER 36
COMPRESS UNIT 42
SPEAKER 32
MICROPHONE 34

FIG. 2

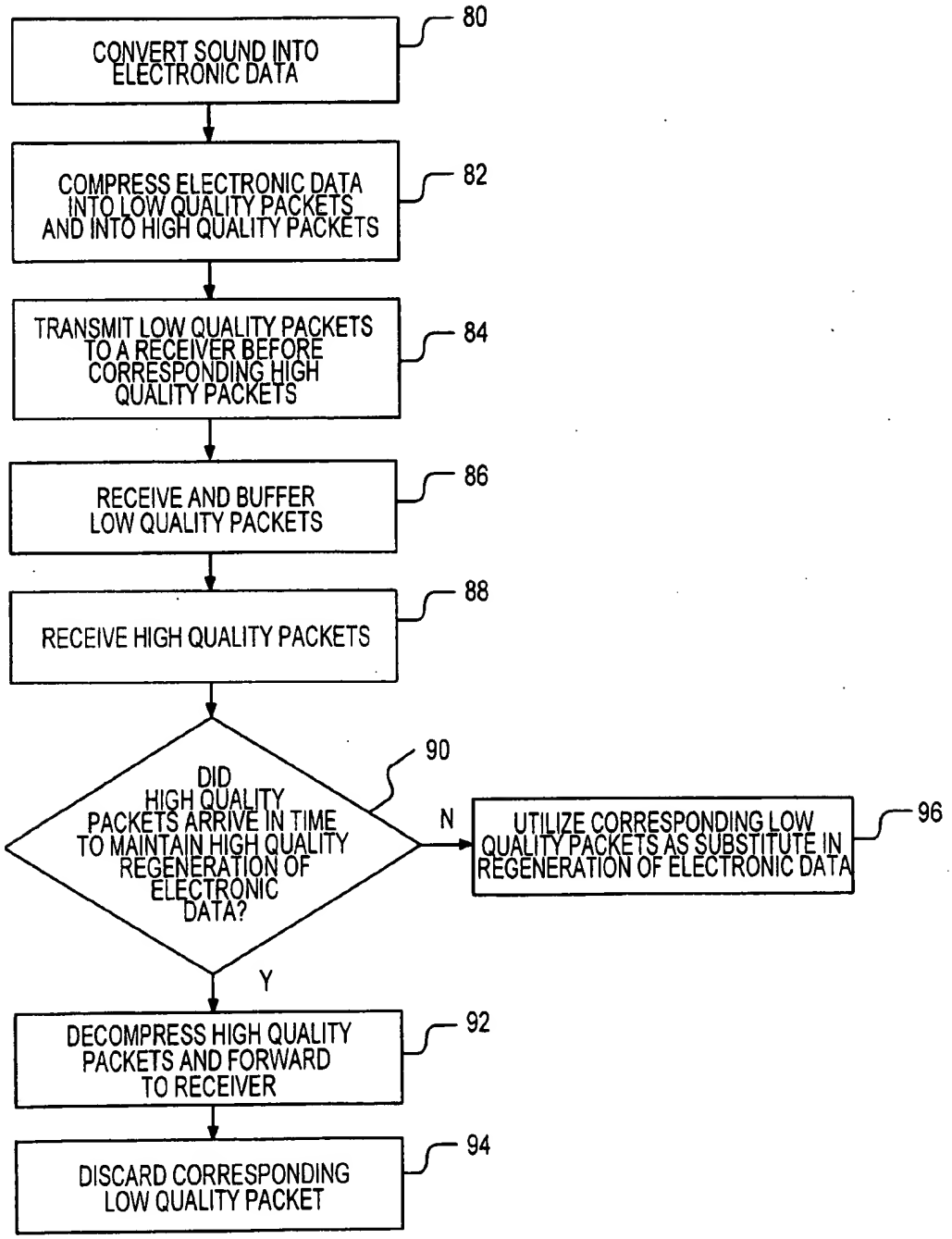

CONVERT SOUND INTO ELECTRONIC DATA
80
COMPRESS ELECTRONIC DATA INTO LOW QUALITY PACKETS AND INTO HIGH QUALITY PACKETS
82
TRANSMIT LOW QUALITY PACKETS TO A RECEIVER BEFORE CORRESPONDING HIGH QUALITY PACKETS
84
RECEIVE AND BUFFER LOW QUALITY PACKETS
86
RECEIVE HIGH QUALITY PACKETS
88
DID HIGH QUALITY PACKETS ARRIVE IN TIME TO MAINTAIN HIGH QUALITY REGENERATION OF ELECTRONIC DATA?
90
N
UTILIZE CORRESPONDING LOW QUALITY PACKETS AS SUBSTITUTE IN REGENERATION OF ELECTRONIC DATA
96
Y
DECOMPRESS HIGH QUALITY PACKETS AND FORWARD TO RECEIVER
92
DISCARD CORRESPONDING LOW QUALITY PACKET
94

FIG. 3

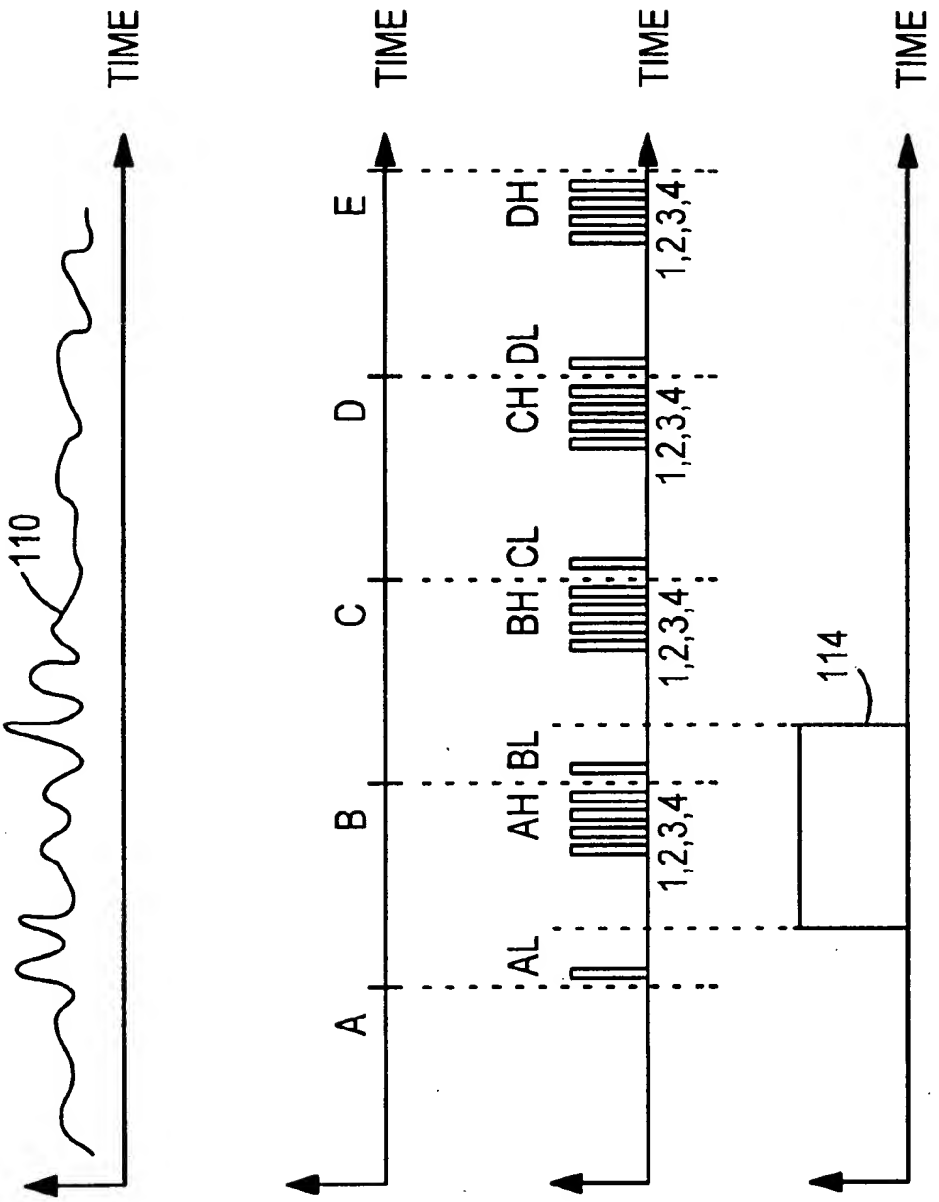


FIG. 4A ANALOG VOICE

FIG. 4B FRAMING CLOCK

FIG. 4C COMPRESSED PACKETS

FIG. 4D TRANSIENT EVENT

METHOD AND APPARATUS FOR PROVIDING A MINIMUM ACCEPTABLE QUALITY OF SERVICE FOR A VOICE CONVERSATION OVER A DATA NETWORK

BACKGROUND OF THE INVENTION

The invention relates generally to data network telephony. More particularly the invention relates to providing a minimum acceptable quality of service for a voice conversation conducted over a data network.

DESCRIPTION OF THE RELATED ART

The publicly switched telephone network (PSTN) is a circuit switched network that has been optimized for real time or synchronous voice communication with a guaranteed quality of service (QoS). When a telephone call is initiated, a circuit is established between the calling party and the called party and the PSTN guarantees QoS by dedicating a full duplex circuit between parties of a telephone conversation. Regardless of whether or not parties are speaking or silent, they are occupying the entire dedicated circuit until the call ends. Since the occupied bandwidth remains constant, the cost of a telephone call on the PSTN is based on distance and time.

On the other hand, typical data networks are packet switched networks that have been used for applications such as e-mail and file transfers where a variable QoS is tolerable. Typical packet switched networks do not dedicate a path between a sender and a receiver and therefore it is harder to guarantee a particular QoS. As data networking technology has improved, the ability to conduct real time conversations over data networks has been developed. By conducting conversations over data networks, access to a PSTN may not be needed and PSTN charges may be avoided. For example, many corporations have extensive enterprise data networks that have untapped capability to carry voice conversations in addition to the data that is being exchanged throughout the network. By channeling voice traffic onto a data network, a corporation may be able to significantly reduce PSTN expenses.

In data networks that transfer data packets according to the popular Internet Protocol (IP), conducting real time voice conversations over a data network is commonly referred to as IP telephony. The term IP telephony is used in the present specification to refer generally to all real time voice conversations conducted through a data network. Besides private data networks such as enterprise networks, IP telephony can be carried out over the global Internet, which also allows users to avoid PSTN expenses beyond the expenses related to Internet access.

Although IP telephony has many advantages, it also has some disadvantages. The main disadvantage of IP telephony is the unpredictable QoS that is provided. The unpredictability is predominantly a result of bandwidth limitations and latency. Bandwidth limitations and latency are often tied together, since when there is insufficient bandwidth in a network to transfer a voice conversation at a desired rate, some packets may be delayed in the transmission to the destination or some packets may be dropped altogether from the transfer because they have taken too much time to be transferred. When packets generated from a voice conversation are delayed or dropped, the quality of the voice conversation carried over the network declines.

One conventional technique used to minimize bandwidth limitations and latency problems in the transmission of voice conversations over a data network is data compression. Data compression allows the amount of voice data from a conversation to be reduced into a smaller number or smaller sized packets for transfer through a network. One problem with compression/decompression algorithms is that the more compressed the data is, the harder it is to decompress the data into an exact replica of the original voice data. As a result, there is a tradeoff between the compression ratio applied to a voice conversation and the quality of the decompressed product.

In order to match the optimal compression ratio to the current bandwidth capacity of a network that is used for voice conversations, systems have been designed that intelligently negotiate and terminate a current compression/decompression algorithm in favor of a more appropriate algorithm depending on current network traffic conditions. An example of a dynamically changing compression/decompression algorithm system is disclosed in U.S. Pat. No. 5,546,395, entitled "Dynamic Selection of Compression Rate for a Voice Compression Algorithm in a Voice Over Data Modem," issued to Sharma et al. (hereafter Sharma). Although Sharma may work well for its intended purpose, the transition between compression/ decompression algorithms typically costs valuable setup time that degrades voice conversation quality.

In view of the current bandwidth limitations and latency involved with IP telephony and the disadvantages of compression/decompression negotiation, what is needed is a method and apparatus for conducting voice conversations over a data network, with sufficient quality and reliability.

SUMMARY OF THE INVENTION

A method and apparatus for transmitting delay-sensitive data over a packet-based network involves converting the delay-sensitive data into two versions for transmission through the network and then using one of the two versions to regenerate the original delay-sensitive data and using the other version to supplement the regeneration of the delay-sensitive data when necessary to compensate for transmission errors or delay that occur during the transmission of the version that was initially used for regeneration. In a preferred embodiment, the delay-sensitive data represents a real time voice conversation and the two versions of the delay-sensitive data include the same segments of the conversation that have been compressed into packets using two different compression algorithms. The first version of the delay-sensitive data is more highly compressed than the second version and because of compression/decompression inefficiencies, the first version provides a lower quality reproduction of the original voice conversation than the second version. Although the first version is of a lower quality, it consumes less bandwidth and has lower latency when transmitted over the network, relative to the more voluminous high quality second version. The low quality version is then used to fill in voice data gaps that are caused when the high quality version does not arrive at its destination on time.

To optimize the quality of the voice conversation, packets from the highly compressed version of the data are sent before packets from the less compressed version, where both sets of packets represent the same segment of the voice conversation. The highly compressed packets are buffered at the receiving end in case they are needed to supplement the less compressed version. Packets from the less compressed version are utilized whenever possible to regenerate the conversation at the receiving end of the transmission, however if packets from the less compressed version are overly

delayed or are dropped, then the corresponding packets from the highly compressed version of the data are supplemented to regenerate the segments of the conversation that would otherwise have been lost or distorted.

The preferred method of the invention is applicable to any packet-based network where the term packet includes data segments referred to as cells, frames, etc. Network protocols applicable to the invention include Internet protocol-based networks, ethernet networks, token ring networks, frame relay networks, and asynchronous transfer mode (ATM) networks.

An IP telephony device designed to enable data transmission in accordance with the invention includes a microphone, a speaker, a converter, memory, a processor, a compression unit, a decompression unit, and a transceiver. The speaker and microphone are conventional devices that are used to send and receive audio information in the frequency range of normal conversation. The converter is a conventional device that converts analog information to digital information and converts digital information to analog information. The converter interfaces with the speaker and the microphone to convert analog voice data from the microphone into digital voice data for the IP telephony device and to convert digital voice data from the IP telephony device to analog voice data for the speakers. The memory is conventional memory that is used to buffer incoming and/or outgoing packets. The processor performs data management functions which include controlling the flow of data between the functional units of the IP telephony device. The compression unit compresses the digital voice data into packets before the packets are sent to a receiving device, and the decompression unit decompresses compressed packets that are received from a sending device. The transceiver includes any conventional device, such as a network interface card or a modem that sends and receives packets of data to and from the data network. Although the components of the IP telephony device are described separately, the functions of the components can be incorporated into a single device or groups of devices other than as explained.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a depiction of a data network that enables voice conversations between IP telephony devices in accordance with the invention.

FIG. **2** is an expanded depiction of two IP telephony devices equipped to enable voice conversation in accordance with the invention.

FIG. **3** is a process flow of a method for conducting a voice conversation between two devices, such as the devices of FIG. **2**, in accordance with the invention.

FIG. **4A** is a time-line of analog voice data.

FIG. **4B** is a time-line related to the time-line of FIG. **4A** that shows framing clocks used for packetizing the analog data of FIG. **4A**.

FIG. **4C** is a time-line related to the time-line of FIGS. **4A** and **4B** that shows one low quality packet to every four high quality packets of the packetized analog voice data from FIG. **4A**.

FIG. **4D** is a time-line related to the time-lines of FIG. **4A–4C** that shows an exemplary transient event that disrupts the transmission of some packetized data.

DETAILED DESCRIPTION

FIG. **1** is a depiction of a data network **10** that connects six IP telephony equipped devices **12, 14, 16, 18, 20,** and **22**. In the depicted embodiment, the network is a corporate enterprise network that transfers packets of information according to IP, although other network protocols are possible. Although depicted as a corporate enterprise network, the network can be any network that is able to carry packetized data, such as the global Internet, where a packet refers to any segment of data including cells and frames. In addition, the data network may transfer data according to protocols such as ethernet, frame relay, or ATM.

The six IP telephony devices **12–22** are preferably personal computers or work stations that are equipped with IP telephony capability and the specifics of the specially equipped devices are discussed in detail below. Although IP telephony equipped computers and work stations are described, the IP telephony devices can alternatively be any IP telephony equipped devices, such as IP specific telephones or hand-held computers. Further, although the IP telephony devices are depicted as physically connected to the network **10**, the IP telephony devices may alternatively have wireless connections to the network.

FIG. **2** is a depiction of two identical IP telephony devices **30** and **50** that are equipped in accordance with the invention and that are connected to the data network **10** of FIG. **1** to enable IP telephony communication in accordance with the invention. The individual functional units of the devices are first described with reference to IP telephony device **30** and then the preferred method of the invention is described with reference to both devices **30** and **50**. The IP telephony device **30** includes a speaker **32**, a unit **42**, a decompression unit **44**, and a transceiver **46**. The IP telephony device **50** has the following corresponding functional units; a speaker **52**, a microphone **54**, a converter **56**, memory **58**, a processor **60**, a compression unit **62**, a decompression unit **64**, and a transceiver **66**.

The speaker **32** and microphone **34** are conventional devices that are used to send and receive audio information in the frequency range of normal conversation. The converter **36** is a conventional device that converts analog information to digital information and converts digital information to analog information. The converter interfaces with the speaker and the microphone to convert analog voice data from the microphone into digital voice data for the IP telephony device and to convert digital voice data from the IP telephony device to analog voice data for the speaker. The memory **38** is conventional memory such as dynamic random access memory (DRAM) that is used to buffer incoming and/or outgoing packets. Although the memory is preferably DRAM, other types of memory may be utilized.

The processor **40** performs data management functions such as directing data flow between the functional units of the IP telephony device. A more detailed explanation of the management functions is provided below in the description of the preferred method of the invention. The compression unit **42** of the device **30** compresses digital voice data into packets before the data is sent to a receiving device, and the decompression unit **44** decompresses previously compressed packets that are received from a sending device. There are many known compression/decompression algorithms that are applicable to compressing different types of data such as voice conversation or video conferencing. Selecting an appropriate compression algorithm depends on factors such as the available bandwidth of the network, delay tolerances, and the desired quality of the decompressed product. For example, a segment of data that is highly compressed data can be transferred over a network using less bandwidth and with less delay but will be of lower quality when decompressed and converted into sound. The

quality of the decompressed voice data degrades because some bits of data are discarded during compression and inaccurately regenerated during decompression. In contrast, data that is less compressed has more delay and requires more bandwidth to transfer, but will have a higher quality when decompressed and converted into sound.

The transceiver **46** includes any conventional device such as a network interface card (NIC) or a modem that sends and receives packets of data to and from the data network. Although the components of the IP telephony device **30** are described separately, the function of the components can be incorporated into the same device or groups of devices. For example, a single component may perform the conversion functions, the compression/decompression functions, and the processor functions and, as will become apparent, the exact layout of the device is not critical to the invention.

FIG. **3** is a process flow diagram used to describe the method of the invention. FIG. **4A** is an exemplary depiction of analog voice data **110** versus time, FIG. **4B** is a depiction of a series of framing clocks that are used to packetize the analog voice data of FIG. **4B**, and FIG. **4C** is a depiction of dual sets of packetized analog voice data from FIG. **4A** using the framing clock of FIG. **4B**, all of which are used to describe the method of the invention. The method of the invention is also described in terms of the IP telephony devices **30** and **50** of FIG. **2** although the devices of FIG. **2** are not the only devices that can be used to carry out the method of the invention.

To begin the method of the invention, a voice conversation between a talker and a listener is initiated by the talker using a respective IP telephony supported microphone **34**. In a first step **80** of the method, the analog audio voice data from the talker, as shown in FIG. **4A**, is converted into electronic voice data by the microphone and then into digital voice data by the converter **36**.

In a next step **82**, each segment, as defined by the framing clocks of FIG. **4B**, of the voice data is compressed into a low quality version of the voice data and into a high quality version of the voice data by the compression unit **42**. As a result of the two compression processes, each original segment of electronic voice data is simultaneously represented in FIG. **4C** by one low quality packet and four high quality packets. The low quality packets are labeled as AL, BL, CL, and DL, where AL is the one low quality packet generated from framing clock A and so on. The high quality packets are labeled as AH, BH, CH, and DH, where AH represents the four high quality packets generated from framing clock A and so on.

In a next step **84**, packets from the low quality version of the electronic voice data are sent to the listener's device **50**, followed by the corresponding packets from the high quality version of the electronic voice data. The low quality version packets are transmitted first because their smaller size enables the low quality packets to pass through the network **10** at a higher transmission rate, thereby providing the listener's device with a more reliable albeit lower quality source of the voice data.

The transmission pattern of dual versions of the compressed data is repeated as the conversation continues, but because data networks handle each packet of data as a unique unit, the packets do not necessarily arrive at the listener's device **50** in the same order and at the same time intervals as originally sent, though it is likely that they do.

In a next step **86**, the low quality packets from the low quality version of the voice data are received at the listener's device **50** before corresponding high quality packets from the high quality version of the voice data. As the low quality version of the voice data is received at the listener's device, the low quality version packets are buffered in a memory **58**. Depending on the processing capability of the device, the low quality version may be initially decompressed prior to buffering, but this is not critical to the invention. In a next step **88**, the high quality version of the voice data is received at the listener's device and the high quality packets are decompressed by the decompression unit **64** according to the specific algorithm that was initially used to compress the high quality version of the electronic voice data.

At decision point **90**, the processor **60** helps to determine whether or not the high quality packets have arrived at the receiving device in time to maintain a minimum quality regeneration of the electronic voice data. A minimum quality regeneration will not be achieved if packets arrive too late or not at all. If the high quality packets arrive at the receiving device in time, at step **92**, the high quality packets are decompressed to regenerate the original electronic voice data and the data is forwarded within the receiving device, so that the original electronic voice data can be transformed into the original audio data and output onto the speaker **52**. If the high quality packets do not arrive at the receiving device **50** in time, at step **96**, the corresponding low quality packets are utilized as substitutes to regenerate the original electronic voice data and subsequently the original audio data. Even though the low quality data may not provide as sharp of a reproduction of the original electronic voice data as the high quality data, at least the segment of electronic voice data can be reproduced and inserted into the flow of data to maintain the continuity of the ongoing voice conversation. In the case where the high quality packets are utilized for a segment of conversation, at step **94** the corresponding low quality packets are preferably discarded from the buffer memory.

The arrivals of the high quality packets and the low quality packets are continuously monitored by the receiving, or listener's, device so that the receiving device can determine when to switch from the high quality packets to the low quality packets. In addition, monitoring allows the receiving device to determine when to transition back from the low quality packets to the high quality packets.

FIG. **4D** is an example depiction of how the method of the invention can operate to enable a reliable real time conversation over a packet based network even when an unpredictable event disrupts network traffic flow. As can be seen, a transient event **114** that disrupts network traffic flow occurs during portions of framing clocks A and B. The transient event is some event that either causes the packets within the time window to be delayed or to be dropped altogether. The transient event in FIG. **4D** causes the high quality packets AH from framing clock A and the low quality packet BL from framing clock B to be lost. In a normal transmission where only one packet sequence is sent, the quality of the voice data output from the receiving device would be degraded, possibly beyond recognition if the packets represent a long enough time period. However, since the low quality packet AL from framing clock A and the high quality packets BH from framing clock B are transmitted successfully without too much delay, the entire sequence of analog voice data can be regenerated by combining the low quality packet from framing clock A with the high quality packets from framing clocks B through D. If a transient event is so large that it covers both sets of compressed data, then the affected segments may be lost in spite of the dual transmission approach.

The method of the invention is especially useful when applied to networks that can provide a specific QoS. For

example, some networks can guarantee bandwidth on a tiered pricing basis where larger amounts of guaranteed bandwidth are more expensive. In this situation, the low quality, low delay, packets can be transferred via the expensive network with guaranteed bandwidth while the high quality, high delay, packets can be transferred via a lower cost network where bandwidth may not be guaranteed. Under this approach, the conversation has high quality when economical bandwidth is available, yet the conversation is also maintained at a lower quality when economical bandwidth is not available. In some cases, both the high quality and the low quality packets are transferred over the exact same route within the network and a resource reservation protocol (RSVP) is used to reserve specified amounts of bandwidth for the designated high quality and low quality traffic.

An advantage of the invention is that less data from a voice conversation is lost because substitute data is sent in parallel with the preferred high quality version of the data. Even though the substitute data is of a lower quality, it is better to fill in the conversation with low quality data than to have voice data gaps which may cause the conversation to be distorted or unintelligible. Another advantage of the invention is that networks providing a guaranteed quality of service can be intelligently utilized to manage IP telephony in a manner that consumes lower cost bandwidth when available while consuming only enough high cost bandwidth to maintain a minimum level of conversation quality. Further, the improved quality and reliability achieved through the invention enable IP telephony to be implemented in corporate enterprise networks, thereby minimizing PSTN expenses.

Although the preferred embodiment of the invention is described with reference to real time voice conversations and IP telephony, the method of the invention is also applicable to other delay-sensitive data transmissions over packet based networks such as video teleconferencing or video broadcasting. The dual mode transmission approach may also apply to other data transfers as bandwidth reservation becomes more widely available.

What is claimed is:

1. A method for ensuring the transmission of delay-sensitive data over a packet based network comprising the steps of:

generating delay-sensitive data at an output device;

converting said delay-sensitive data into a first version for transmission to an input device;

converting said delay-sensitive data into a second version for transmission to said input device;

transmitting said first version and second version of said delay-sensitive data to said input device;

regenerating said delay-sensitive data from said second version, thereby forming regenerated data at said input device; and

supplementing said regenerated data with data regenerated from said first version to compensate for transmission delays that occur during said transmission of said second version.

2. The method of claim 1 wherein said step of converting said delay-sensitive data into said first version includes a step of compressing said delay-sensitive data into a first compressed version using a first compression algorithm.

3. The method of claim 2 wherein said step of converting said delay-sensitive data into said second version includes a step of compressing said delay-sensitive data into a second compressed version using a second compression algorithm.

4. The method of claim 3 wherein said first compression algorithm has a higher compression ratio than said second compression algorithm.

5. The method of claim 4 wherein said step of transmitting includes a step of transmitting said first version of said delay-sensitive data to said input device before transmitting said second version to said input device.

6. The method of claim 5 further including a step of buffering said first version at said input device for use in said step of supplementing.

7. The method of claim 5 further including a step of maintaining parallel connections between said output device and said input device for transmitting said first version and said second version.

8. The method of claim 1 wherein said step of generating said delay-sensitive data includes processing one of a voice conversation and a video stream.

9. An apparatus for enabling a voice conversation over a packet based data network comprising:

means for generating electronic voice data;

means for compressing said electronic voice data into a first compressed version;

means for compressing said electronic voice data into a second compressed version that is less compressed than said first version;

means for transmitting said first compressed version over said data network before said second compressed version;

means for regenerating said electronic voice data from said second compressed version; and

means for supplementing said regeneration of said electronic voice data from said second compressed version with said first compressed version when said second compressed version does not arrive from said data network in a manner to maintain a minimum level of data transmission quality.

10. The apparatus of claim 9 further including a microphone and a speaker.

11. The apparatus of claim 10 wherein said means for supplementing includes memory for buffering said first compressed version of said electronic voice data.

12. The apparatus of claim 11 further including a means for generating audio voice data from said regenerated electronic voice data.

13. A method for improving the quality of a delay-sensitive data transmission between an output device and an input device through a data network comprising the steps of:

generating a segment of delay-sensitive data;

compressing said segment of delay-sensitive data into a first compressed version according to a first compression algorithm;

compressing said segment of delay-sensitive data into a second compressed version according to a second compression algorithm wherein said first version is more highly compressed than said second version;

transmitting said first compressed version of said segment of delay-sensitive data from said output device to said input device;

transmitting said second compressed version of said segment of delay-sensitive data from said output device to said input device;

utilizing said second compressed version in a first attempt to regenerate said segment of delay-sensitive data; and

utilizing said first compressed version in a second attempt to regenerate said segment of delay-sensitive data when

said first attempt is unsuccessful in achieving a minimum level of data transmission quality.

14. The method of claim 13 wherein said step of transmitting said first compressed version is performed before said step of transmitting said second compressed version.

15. The method of claim 14 further comprising a step of monitoring the arrival of said second compressed version at said input device to determine when to utilize said first version in place of said second version.

16. The method of claim 14 wherein said step of utilizing said first compressed version includes a step of decompressing said first compressed version in order to regenerate said segment of delay-sensitive data and said step of utilizing said second compression version includes a step of decompressing said second compressed version in order to regenerate said segment of delay-sensitive data.

17. The method of claim 14 wherein said step of transmitting said first compressed version is a step of utilizing a relatively high cost transmission mode and wherein said step of transmitting said second compressed version is a step of utilizing a relatively low cost transmission mode.

18. The method of claim 13 wherein said step of generating a segment of delay-sensitive data includes a step of transforming sound waves into electronic data.

19. The method of claim 13 wherein said step of generating a segment of delay-sensitive data includes a step of generating video data.

20. The method of claim 13 wherein said steps of transmitting include preceding steps of formatting said first and second compressed versions according to the Internet Protocol (IP).

* * * * *

US006363065B1

(12) **United States Patent**
Thornton et al.

(10) **Patent No.: US 6,363,065 B1**
(45) **Date of Patent: Mar. 26, 2002**

(54) **OKAPPARATUS FOR A VOICE OVER IP (VOIP) TELEPHONY GATEWAY AND METHODS FOR USE THEREIN**

(75) Inventors: **Timothy R. Thornton**, Brick; **Rajiv Bhatia**, Marlboro; **Ki Choon Suh**, Middletown, all of NJ (US)

(73) Assignee: **Quintum Technologies, Inc.**, Eatontown, NJ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/437,644**

(22) Filed: **Nov. 10, 1999**

(51) **Int. Cl.**[7] **H04L 12/66**; H04L 12/28; H04M 7/00

(52) **U.S. Cl.** **370/352**; 370/401; 379/221.01

(58) **Field of Search** 370/237, 242, 370/244, 251, 252, 353, 355, 400, 401, 410, 406, 356; 379/16, 17, 221.9, 220.01, 225

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,838,683	A	11/1998	Corley et al.
5,875,234	A	2/1999	Clayton et al.

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

EP	0 848 560	6/1998
EP	0 848 560 A2	6/1998

(List continued on next page.)

OTHER PUBLICATIONS

A. Cray, "IP PBXs: Open Questions, Net managers trying to escape the tyranny of traditional PBX vendors have a cheaper option. But can they really trust the IP alternative?", Data Communications, Mar. 1999, pp. 69–70, 72, 74, 76, 78, 80, 82, 83.

"Star Vox: Solutions: Multi–site Network Telephony Solution", downloaded on Jul. 20, 1999 from http://www.star-vox.com/solutions/sol$_{13}$ msnt.html.

"Star Vox: Market Trends: Migration of Voice to Data Network", downloaded on Jul. 20, 1999 from http://www.starvox.com/trends/tds$_{13}$ mvdn.html.

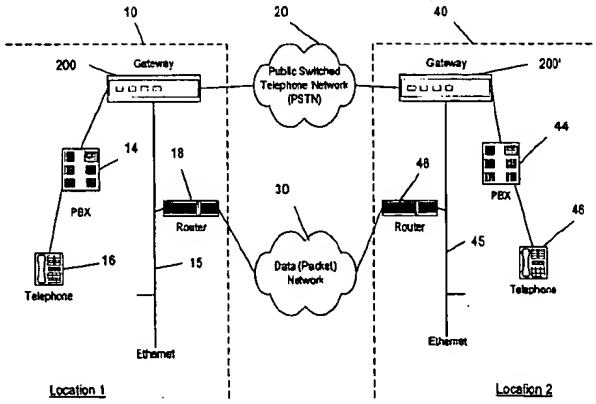
(List continued on next page.)

Primary Examiner—Alpus H. Hsu
Assistant Examiner—Duc Ho
(74) *Attorney, Agent, or Firm*—Michaelson & Wallace; Peter L. Michaelson

(57) **ABSTRACT**

Apparatus, and accompanying methods for use therein, for a telephony gateway intended for use, e.g., paired use, at opposite ends of a data network connection, in conjunction with at each end, e.g., a private branch exchange (PBX) for automatically routing telephone calls, e.g., voice, data and facsimile, between two peer PBXs over either a public switched telephone network (PSTN) or a data network, based on, among other aspects, cost considerations for handling each such call and called directory numbers, monitoring quality of service (QoS) then provided through the data network and switching ("auto-switching") such calls back and forth between the PSTN and the data network, as needed, in response to dynamic changes in the QoS such that the call is carried over a connection then providing a sufficient QoS. To support auto-switching, the apparatus embeds, using call independent signaling, certain call-specific information, as non-standard data, within various conventional H.323 messages that transit between the paired gateways. Furthermore, for added local redundancy, this apparatus utilizes peered border elements within an H.323 administrative domain.

90 Claims, 26 Drawing Sheets



5

U.S. PATENT DOCUMENTS

5,933,490	A	*	8/1999	White et al.	379/221
6,026,087	A	*	2/2000	Mirashrafi et al.	370/389
6,064,653	A	*	5/2000	Farris	370/237
6,185,204	B1	*	2/2001	Voit	370/352

FOREIGN PATENT DOCUMENTS

WO	WO 99/05590		2/1999	
WO	WO-99/05590	*	2/1999	G06F/3/00
WO	WO 99/056590		2/1999	

OTHER PUBLICATIONS

"Nortel Networks Introduces Next Generation Internet Telephony for Enterprises, Brings Together the Worlds of Telephony and Data", downloaded on Jul. 20, 1999 from http://www.nortelnetworks.com/corporate/.../6_{13} 8_9999393Enterprise_Internet_Telephonyt.html.

"Nortel Networks: CALA Press Room: Key Corporate News", downloaded on Jul. 20, 1999 from http://www.nortelnetworks.com/corporate/global/cala/press/keynews/entjune99_eng.html.

"Nortel Networks: News & Events: Inca: Internet Telephony Portfolio", downloaded on Jul. 20, 1999 from http://www.nortelnetworks.com/corporate/events/incal/ internet_telephony_portfolio. html.

"NetPhone IPBX, Intranet PBX and IP Telephony Gateway, Features and Benefits".

"NetPhone, Communications Servers, The Standard for Small to Medium Sized Businesses".

"NetPhone, NetPhone Connect, IP Telephony Gateway".

"Cisco—Cisco IP Telephony", downloaded Mar. 31, 1999 from http://www/cisco.com/warp/public/744/.

"Cisco—Cisco IP Telephony: Product Literature", downloaded Mar. 31, 1999 from http://www.cisco.com/warp/public/744/literature.shtml.

"Cisco IP Telephone Model 12 SP+", downloaded Mar. 31, 1999 from http://www.cisco.com/warp/public/744/ipphone/sp12p_ds.htm.

"Cisco IP Telephone Model 30 VIP", downloaded Mar. 31, 1999 from http://www.cisco.com/warp/public/744/ipphone/phone_ds.htm.

"Cisco CallManager 2.2 and Integrated Voice Applications", downloaded Mar. 31, 1999 from http://www.cisco.com/warp/public/744/callmgr/callm_ds.htm.

"Cisco Access Analog Station Gateways—Models AT–2,4, 8", downloaded Mar. 31, 1999 from http://www.cisco.com/warp/public/744/gateway/as248_ds.htm.

"Cisco Access Analog Trunk Gateways—Models AT–2,4, 8", downloaded Mar. 31, 1999 from http://www.cisco,com/warp/public/744/gateway/at248_ds.htm.

"Cisco Access Digital Gateway (T1 Rate)—Model DT–24", downloaded Mar. 31, 1999 from http://www.cisco.com/warp/public/744/gateway/dt24_ds.htm. "Enterprise IP Packet Telephony Solutions Guide", downloaded Mar. 31, 1999 from http://www.cisco,com/warp/public/744/iptel_wp.htm.

"Enterprise IP Packet Telephone Solutions Guide", downloaded Mar. 31, 1999 from http://www.cisco.com/warp/public/744/iptel_wp.htm.

"Control Protocol for Multimedia Communication, Series H: Audiovisual and Multimedia Systems Infrastructure of Audiovisual Services—Communication Procedures", ITU–T Recommendation H.245, ITU–T Telecommunication Standardization Sector of ITU, International Telecommunication Union, Jul. 1997.

"Packet–based Multimedia Communications Systems, Series H: Audiovisual and Multimedia Systems Infrastructure of Audiovisual Services—Systems and Terminal Equipment for Audiovisual Services", ITU–Recommendation H.245, ITU–T Telecommunication Standardization Sector of ITU, International Telecommunication Union, Version 3, May 1989.

Communication Between Administrative Domains, H.225 Annex G Draft Decision, International Telecommunication Union, Santiago, Chile, May 17–28, 1999.

Draft H.225.0 Version 3, International Telecommunication Union, Santiago, Chile, May 18–28, 1999.

* cited by examiner

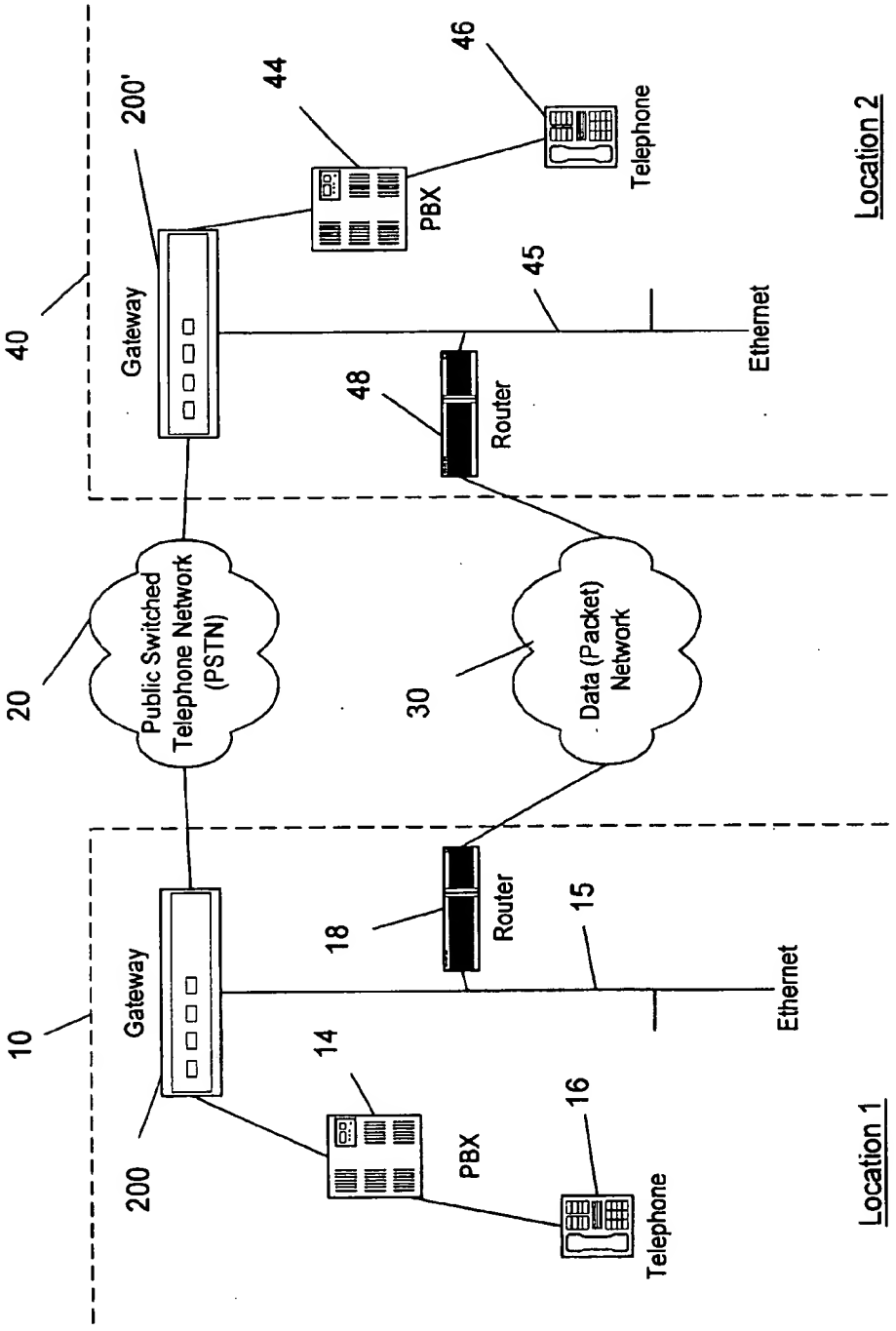

Location 2
Location 1
40
10
200'
200
Gateway
Gateway
44
PBX
46
Telephone
45
Ethernet
48
Router
18
Router
15
Ethernet
14
PBX
16
Telephone
20
Public Switched Telephone Network (PSTN)
30
Data (Packet) Network
5

FIG. 1

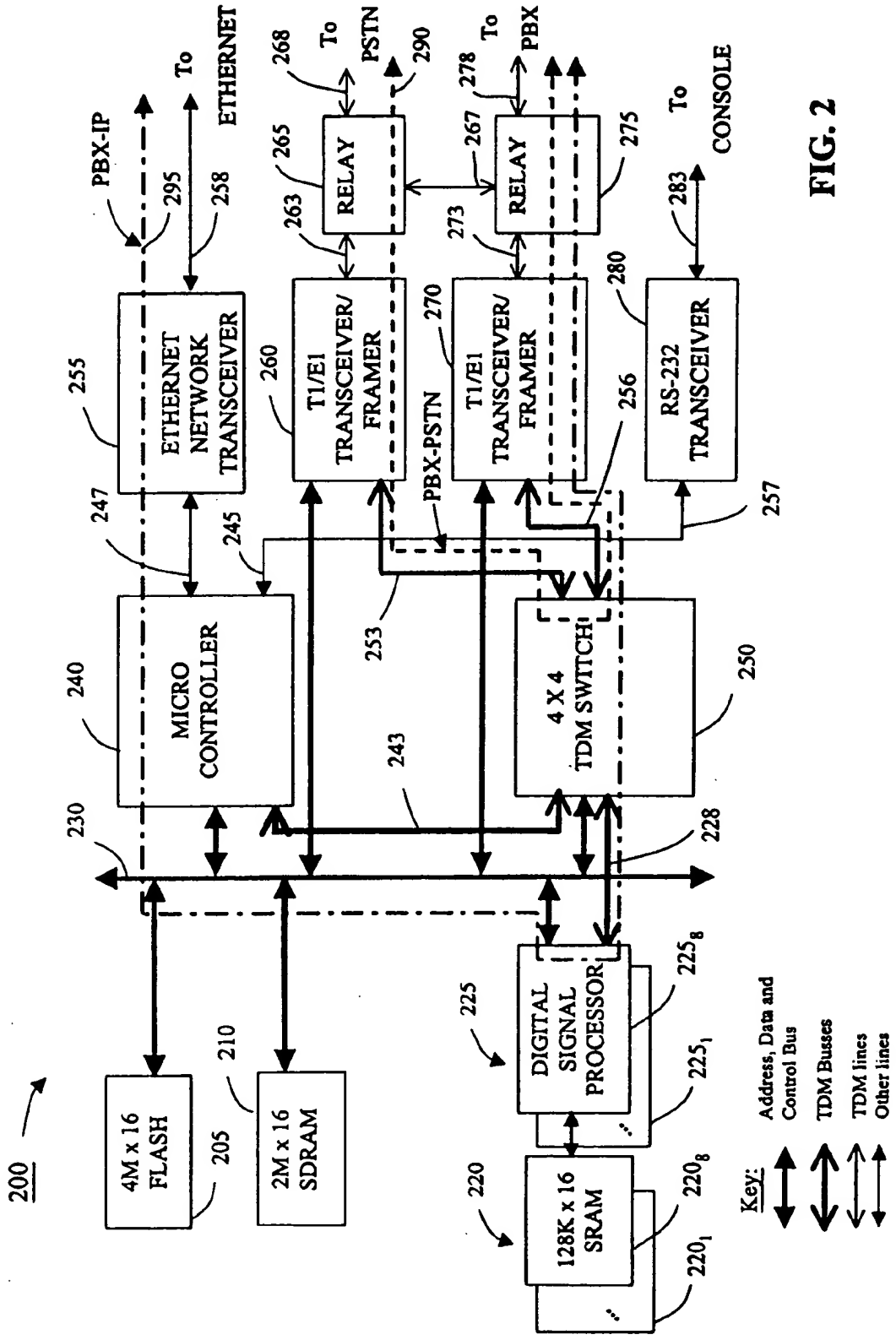

200
4M x 16 FLASH
205
2M x 16 SDRAM
210
MICRO CONTROLLER
240
ETHERNET NETWORK TRANSCEIVER
255
T1/E1 TRANSCEIVER/ FRAMER
260
T1/E1 TRANSCEIVER/ FRAMER
270
RS-232 TRANSCEIVER
280
4 X 4 TDM SWITCH
250
DIGITAL SIGNAL PROCESSOR
225
128K x 16 SRAM
220
RELAY
265
RELAY
275
PBX-IP
295
To ETHERNET
258
To PSTN
268
290
To PBX
278
To CONSOLE
283
PBX-PSTN
253
230
228
243
247
245
257
256
263
273
Key:
Address, Data and Control Bus
TDM Busses
TDM lines
Other lines

FIG. 2

FIG. 3

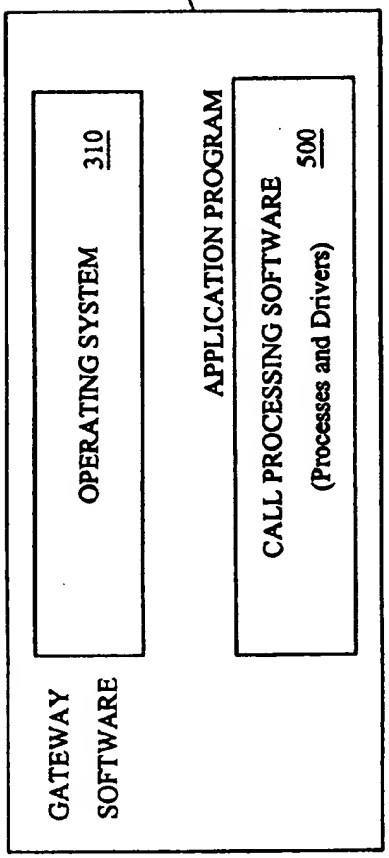


FIG. 6

600

Process(es)	Relative Priority	
TCP/IP 535, CM 505	255	Highest execution priority
VPH 517	200	This process is in the data path for packetized voice transfer between the T1/E1 trunks (via the DSPs) and Ethernet. Hence, a high execution priority is needed to minimize delay associated with VoIP packets.
GK 700, BE 900	150	Gatekeeper and border element processes
CH 560, T1AB 575, P.323 553, Q.931 577, Q.921 572	100	These processes are related to call control and signaling.
Others	50	These processes are mostly related to configuration, network management, etc.
IDLE (502)	10	This priority should be lower than that for all other processes but higher than O/S internal idle process.

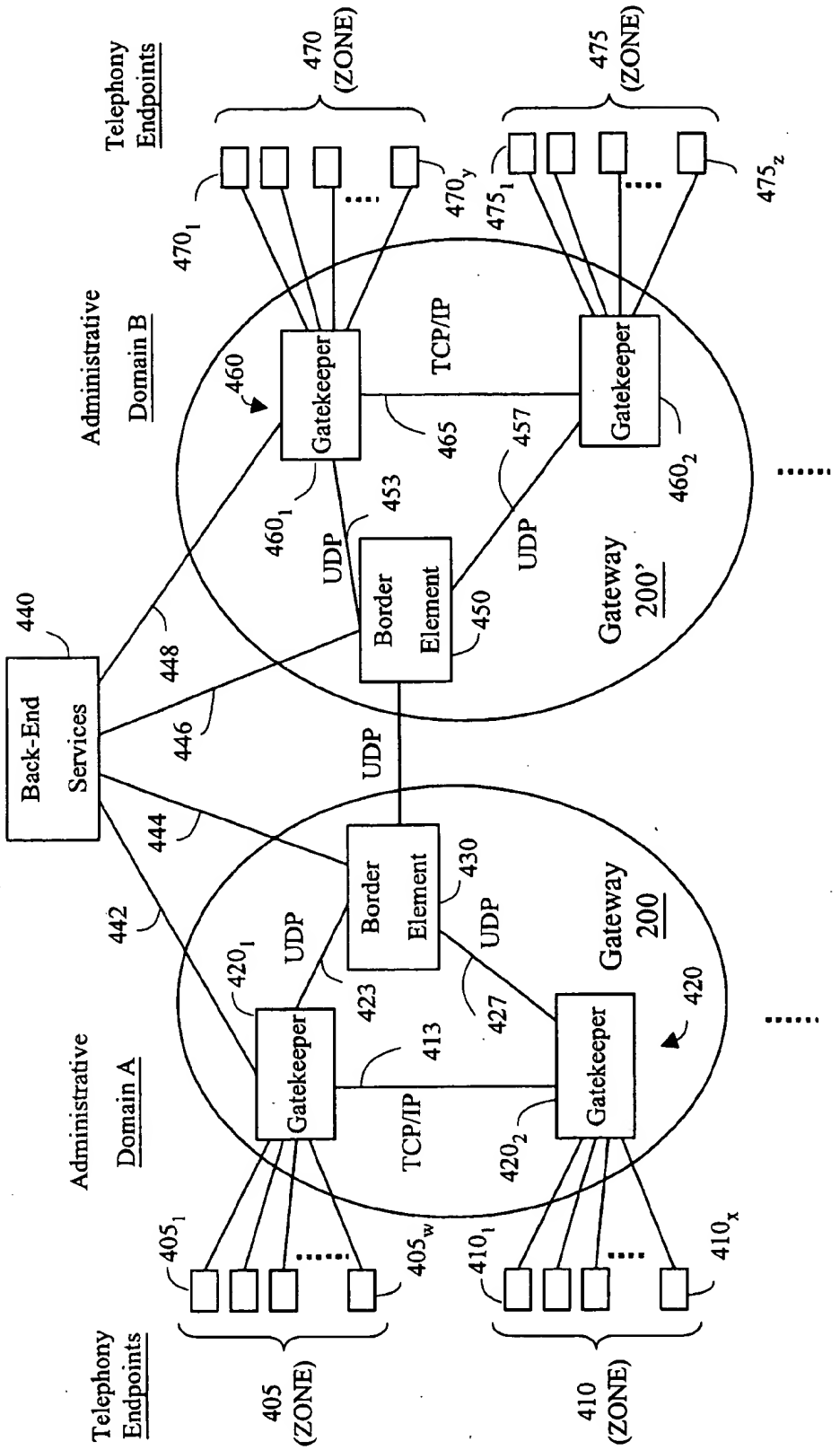

Telephony Endpoints
470 (ZONE)
475 (ZONE)
470_1
470_y
475_1
475_z
Administrative Domain B
460
Gatekeeper
TCP/IP
465
457
Gatekeeper
460_1
453
460_2
UDP
UDP
Border Element
450
Gateway 200'
440
Back-End Services
448
446
UDP
444
442
Border Element
430
Gateway 200
420_1
UDP
UDP
423
427
420
Administrative Domain A
Gatekeeper
413
Gatekeeper
TCP/IP
420_2
405_1
405_w
410_1
410_x
Telephony Endpoints
405 (ZONE)
410 (ZONE)

FIG. 4A

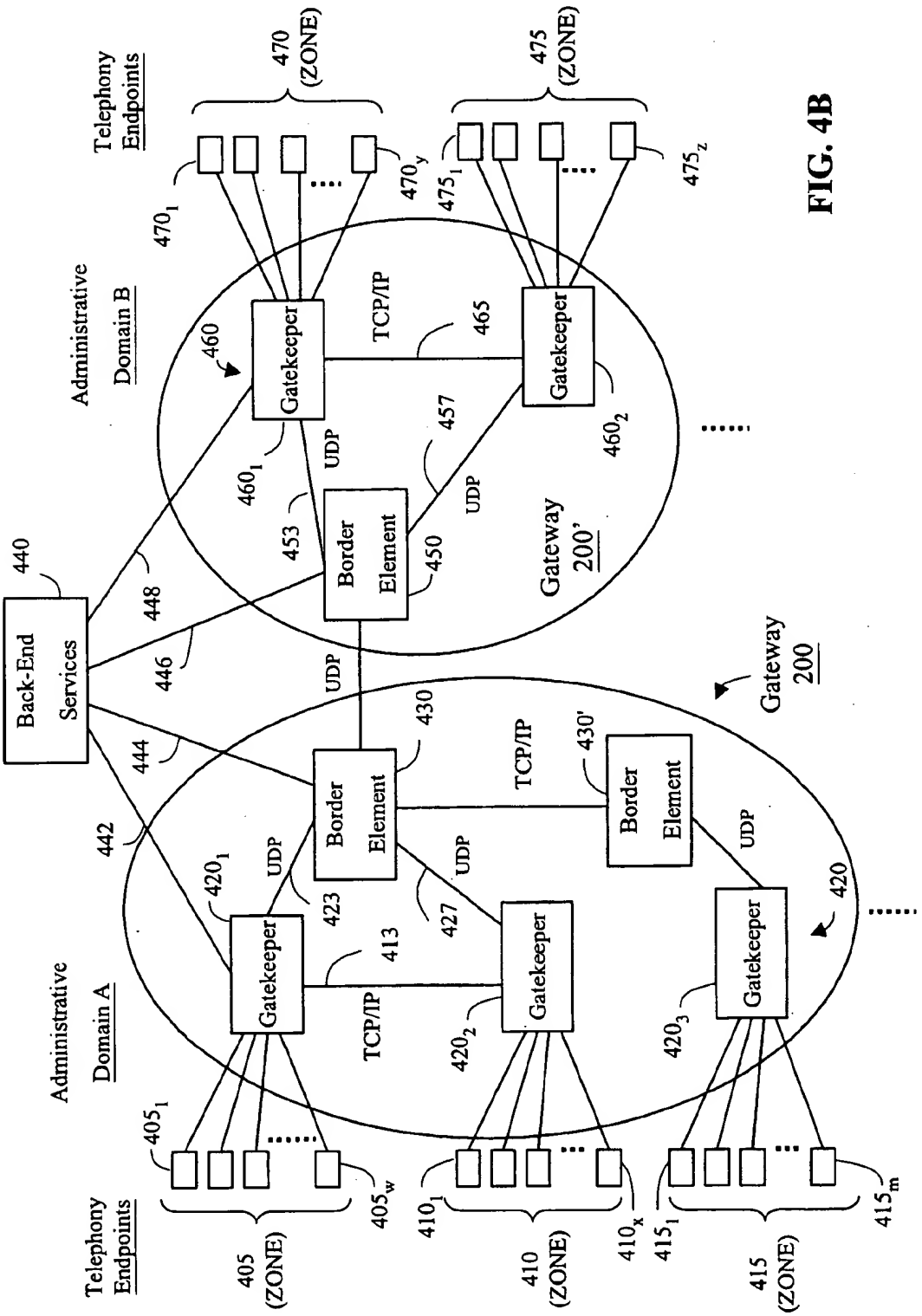
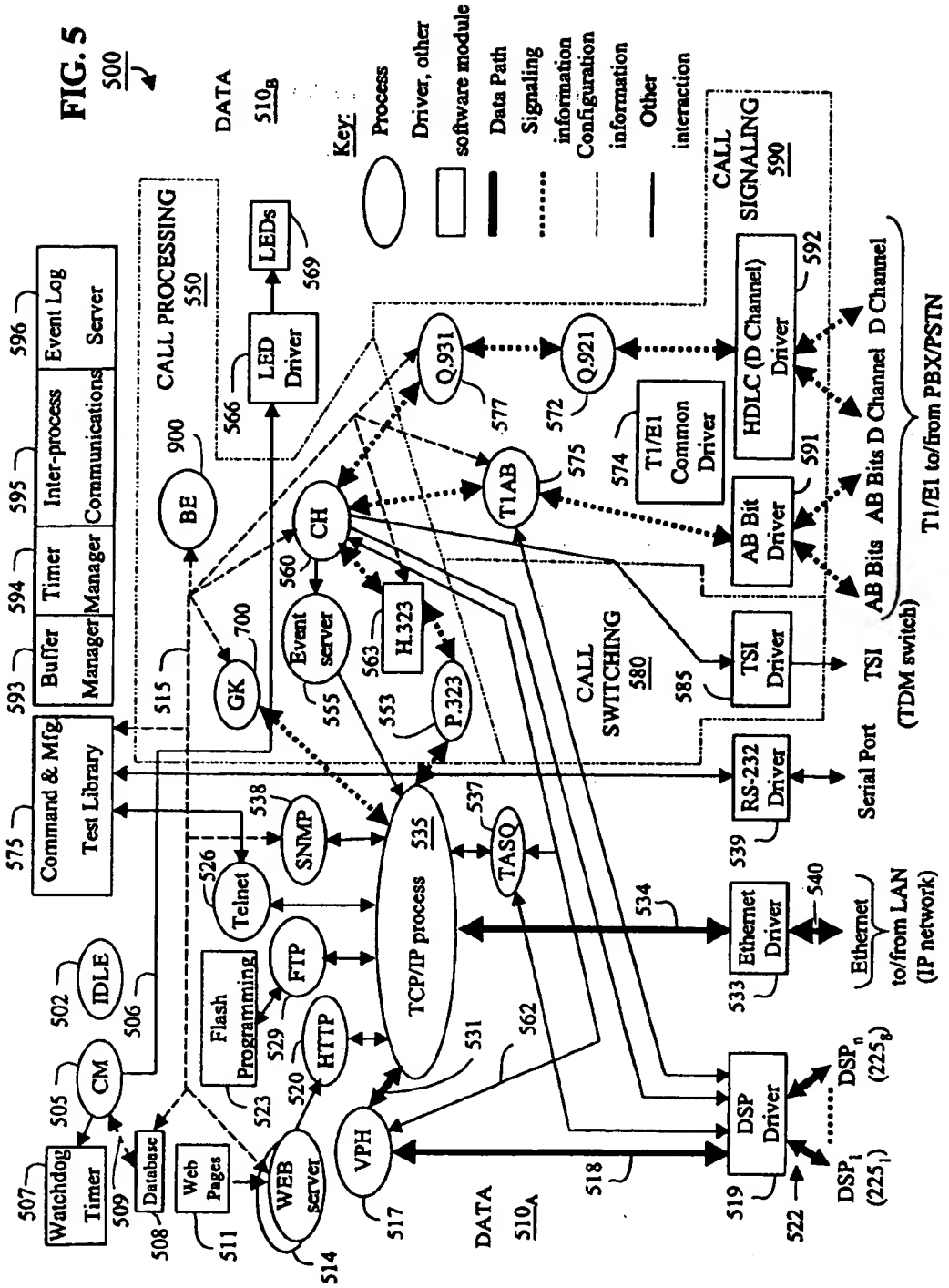

Telephony Endpoints
470 (ZONE)
475 (ZONE)
470_1
470_y
475_1
475_z
Administrative Domain B
460
Gatekeeper
TCP/IP
465
Gatekeeper
460_1
UDP
457
460_2
453
Border Element
450
UDP
Gateway 200'
440
Back-End Services
448
446
UDP
Gateway 200
444
430
Border Element
TCP/IP
430'
Border Element
442
420_1
UDP
UDP
UDP
423
427
413
Gatekeeper
Gatekeeper
Gatekeeper
420
Administrative Domain A
TCP/IP
420_2
420_3
405_1
405_w
410_1
410_x
415_1
415_m
Telephony Endpoints
405 (ZONE)
410 (ZONE)
415 (ZONE)

FIG. 4B

FIG. 5


500
Key:
Process
Driver, other software module
Data Path
Signaling information
Configuration information
Other interaction
CALL PROCESSING 550
CALL SIGNALING 590
CALL SWITCHING 580
DATA 510B
DATA 510A
Command & Mfg Test Library 575
Buffer Manager 593
Timer Manager 594
Inter-process Communications 595
Event Log Server 596
BE 900
CH 560
GK 700
Event server 555
H.323 563
P.323 553
LED Driver 566
LEDs 569
Q.931 577
Q.921 572
T1AB 575
T1/E1 Common Driver 574
HDLC (D Channel) Driver 592
AB Bit Driver 591
TSI Driver 585
RS-232 Driver 539
Serial Port
AB Bits
D Channel
T1/E1 to/from PBX/PSTN (TDM switch)
TSI
TCP/IP process 535
TASQ 537
SNMP 538
Telnet 526
FTP
HTTP 529
Flash Programming 523
IDLE 502
CM 505
506
Watchdog Timer 507
509
Database 508
Web Pages 511
WEB server 514
VPH 517
520
531
562
Ethernet Driver 533
534
540
Ethernet to/from LAN (IP network)
DSP Driver 519
518
522
DSP1 (225)
DSPn (225n)

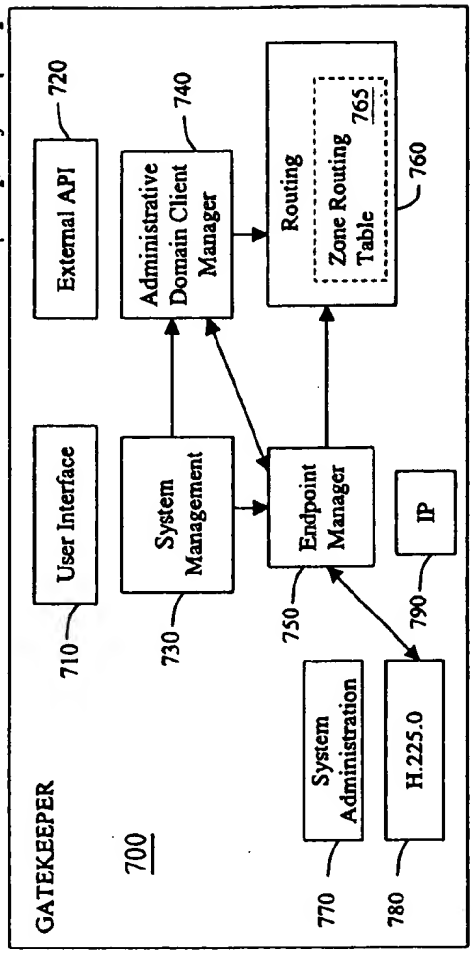

420_1, 420_2,420_3 /460_1,460_2
GATEKEEPER
700
User Interface
710
External API
720
System Management
730
Administrative Domain Client Manager
740
Endpoint Manager
750
Routing
Zone Routing Table
765
760
IP
790
System Administration
770
H.225.0
780

FIG. 7

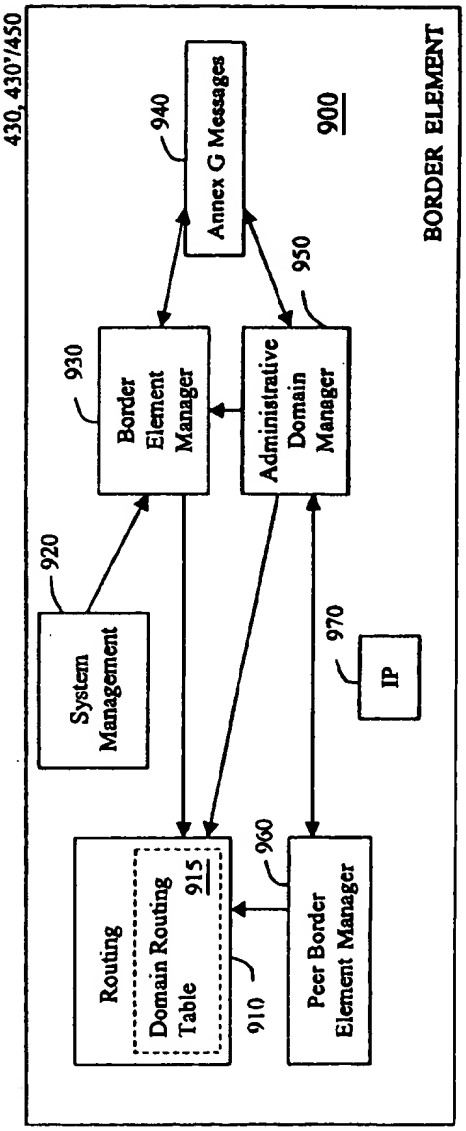

430, 430'/450
BORDER ELEMENT
900
Annex G Messages
940
Border Element Manager
930
Administrative Domain Manager
950
System Management
920
IP
970
Routing
Domain Routing Table
915
910
Peer Border Element Manager
960

FIG. 9

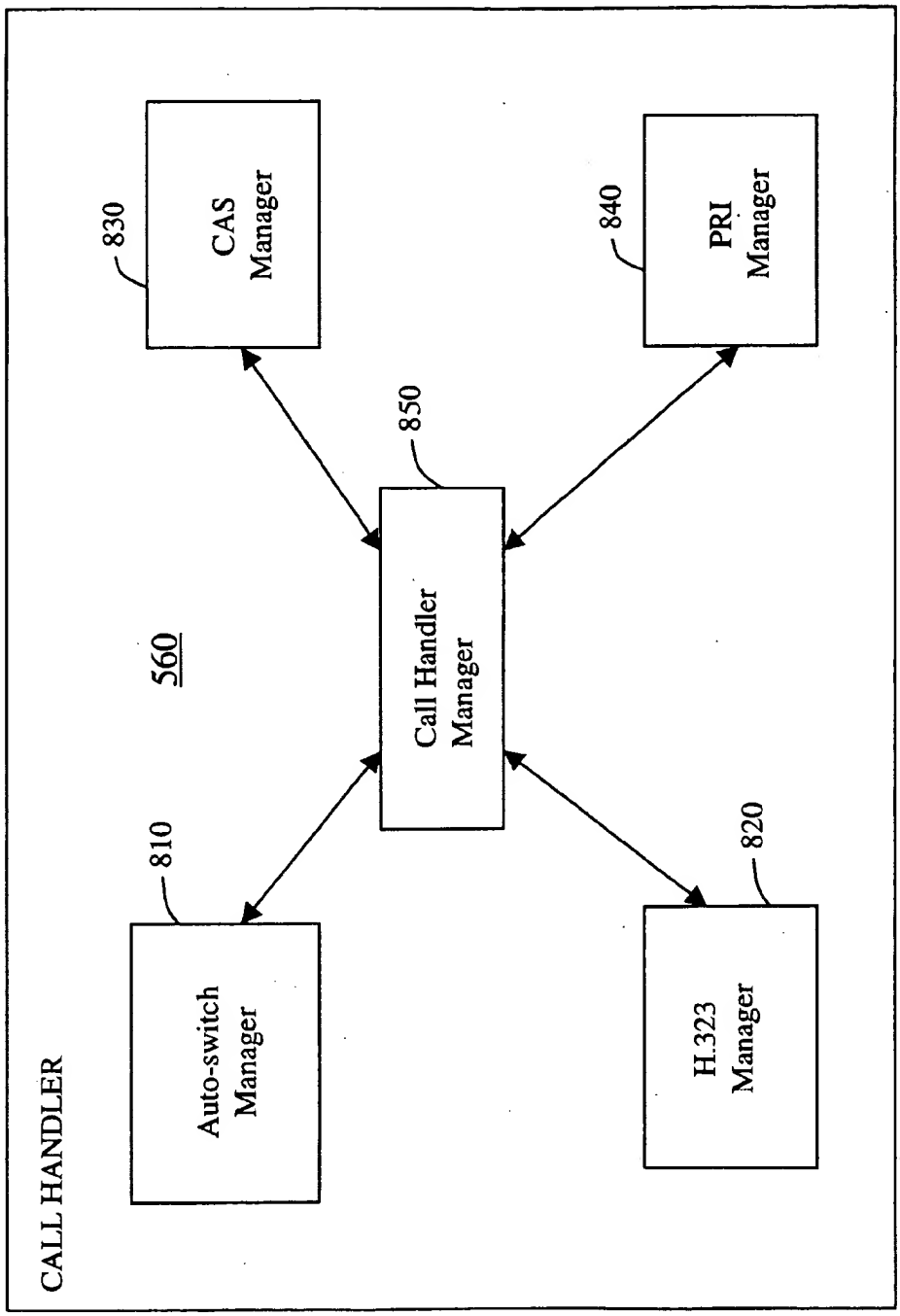

CALL HANDLER
560
Call Handler Manager
850
CAS Manager
830
PRI Manager
840
Auto-switch Manager
810
H.323 Manager
820

FIG. 8

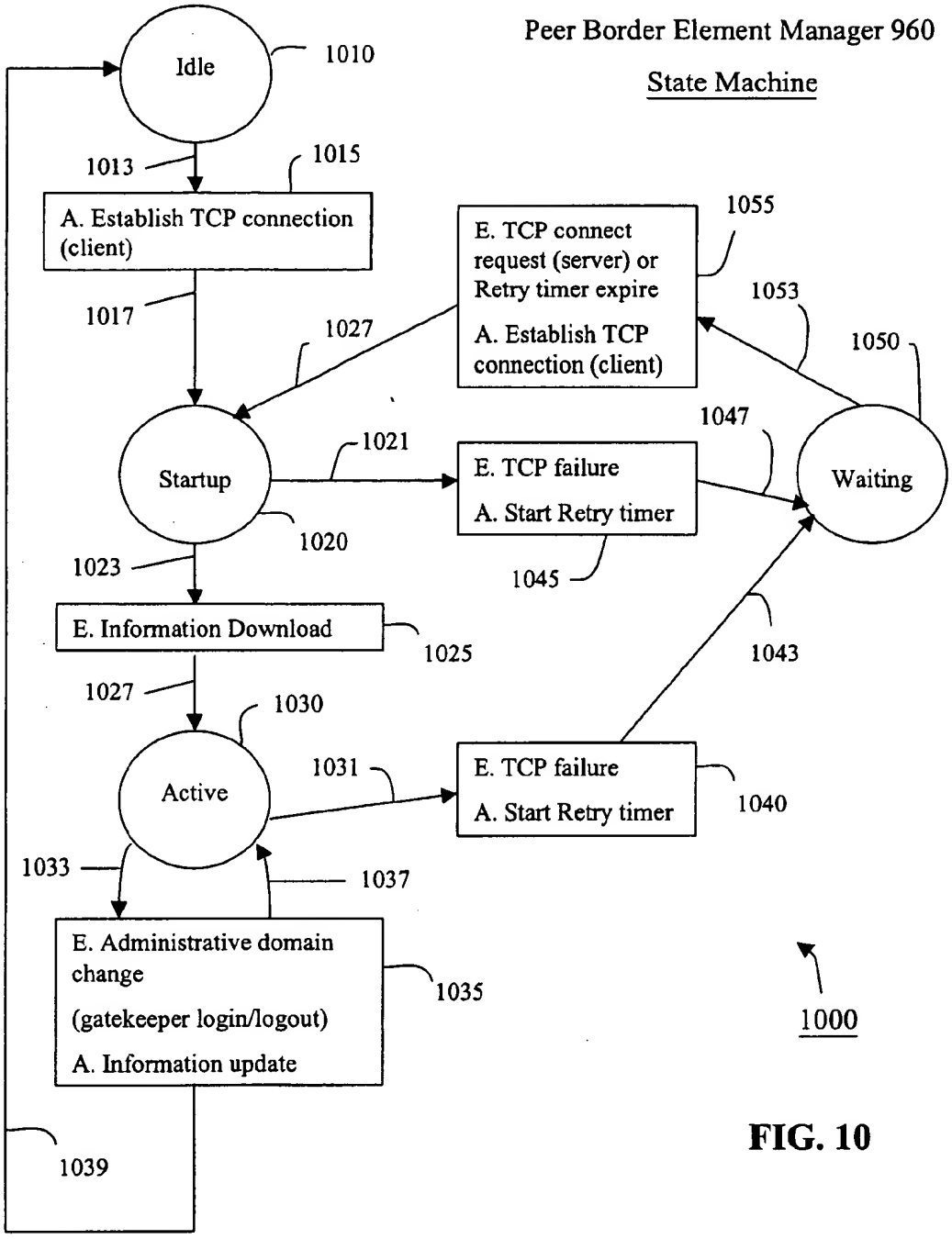

Peer Border Element Manager 960
State Machine
Idle
1010
1013
1015
A. Establish TCP connection (client)
1017
1027
E. TCP connect request (server) or Retry timer expire
A. Establish TCP connection (client)
1055
1053
1050
1047
Startup
1021
E. TCP failure
A. Start Retry timer
Waiting
1020
1045
1023
E. Information Download
1025
1043
1027
1030
Active
1031
E. TCP failure
A. Start Retry timer
1040
1033
1037
E. Administrative domain change
(gatekeeper login/logout)
A. Information update
1035
1000
1039

FIG. 10

FIG. 11

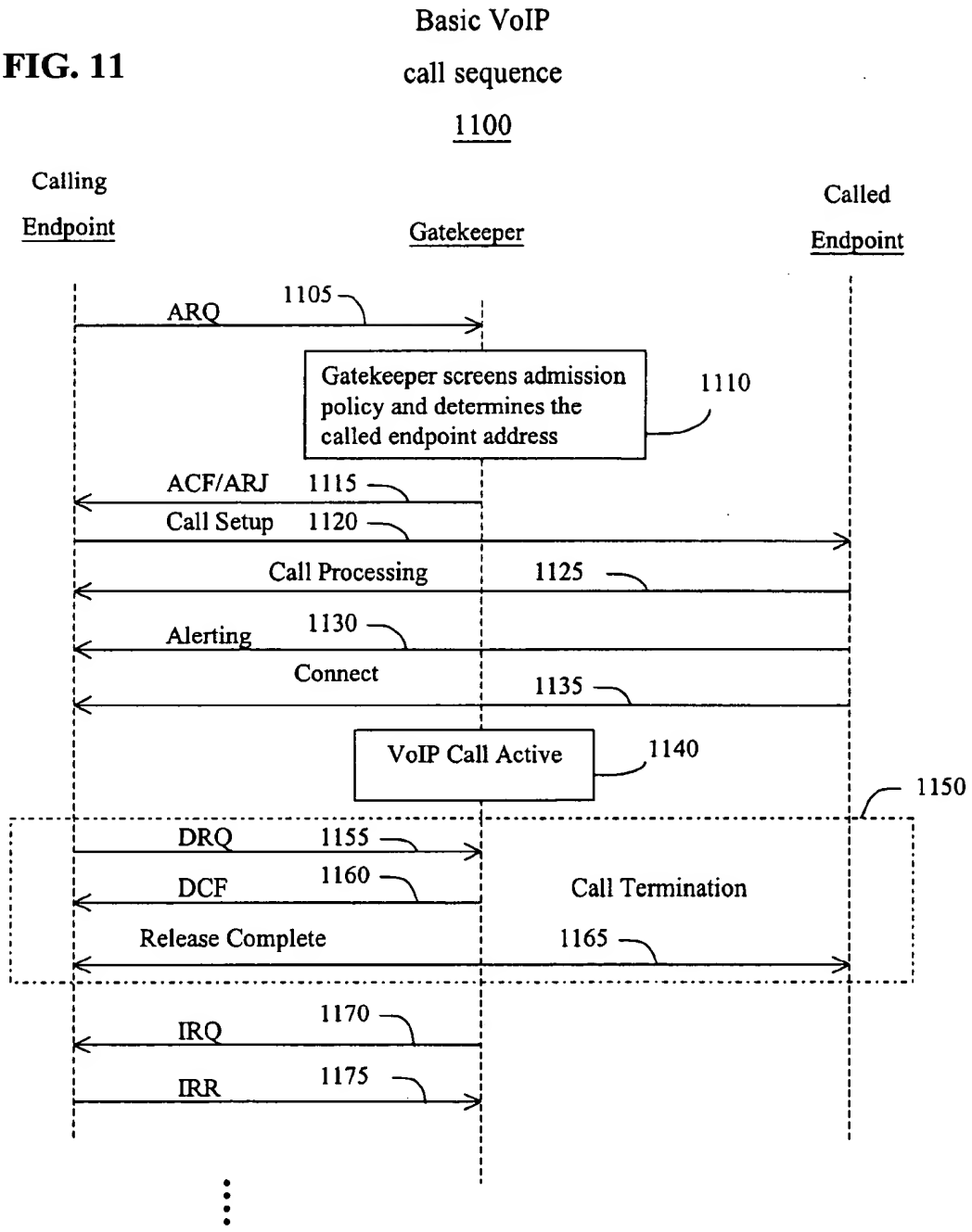
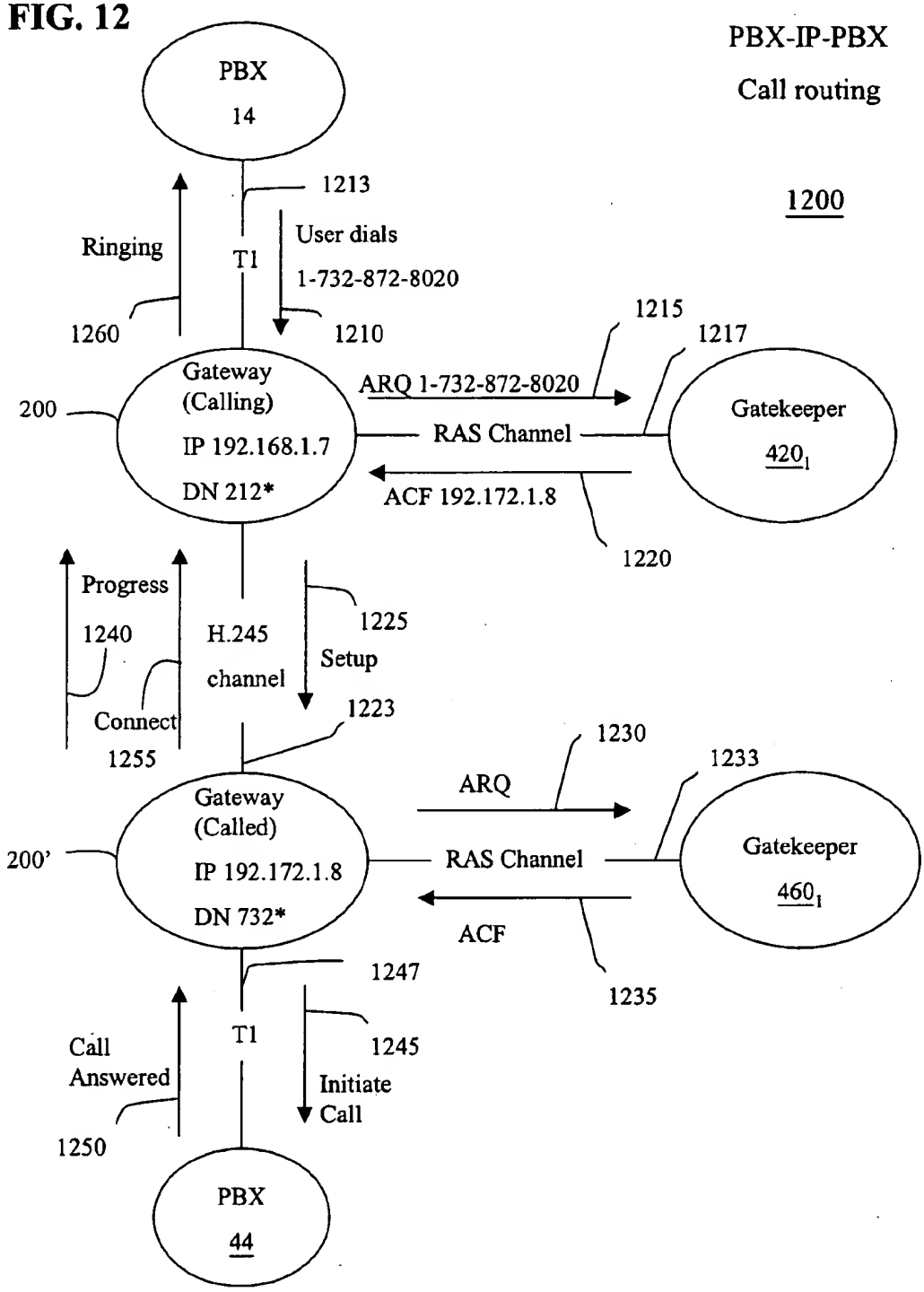

Basic VoIP
call sequence
1100
Calling
Endpoint
Gatekeeper
Called
Endpoint
ARQ
1105
Gatekeeper screens admission policy and determines the called endpoint address
1110
ACF/ARJ
1115
Call Setup
1120
Call Processing
1125
Alerting
1130
Connect
1135
VoIP Call Active
1140
1150
DRQ
1155
DCF
1160
Call Termination
Release Complete
1165
IRQ
1170
IRR
1175

FIG. 12


PBX-IP-PBX
Call routing
1200
PBX
14
1213
Ringing
T1
User dials
1-732-872-8020
1260
1210
1215
1217
Gateway
(Calling)
IP 192.168.1.7
DN 212*
200
ARQ 1-732-872-8020
RAS Channel
ACF 192.172.1.8
Gatekeeper
420$_1$
1220
Progress
1240
H.245
channel
1225
Setup
Connect
1255
1223
1230
1233
Gateway
(Called)
IP 192.172.1.8
DN 732*
200'
ARQ
RAS Channel
ACF
Gatekeeper
460$_1$
1235
1247
T1
1245
Call
Answered
Initiate
Call
1250
PBX
44

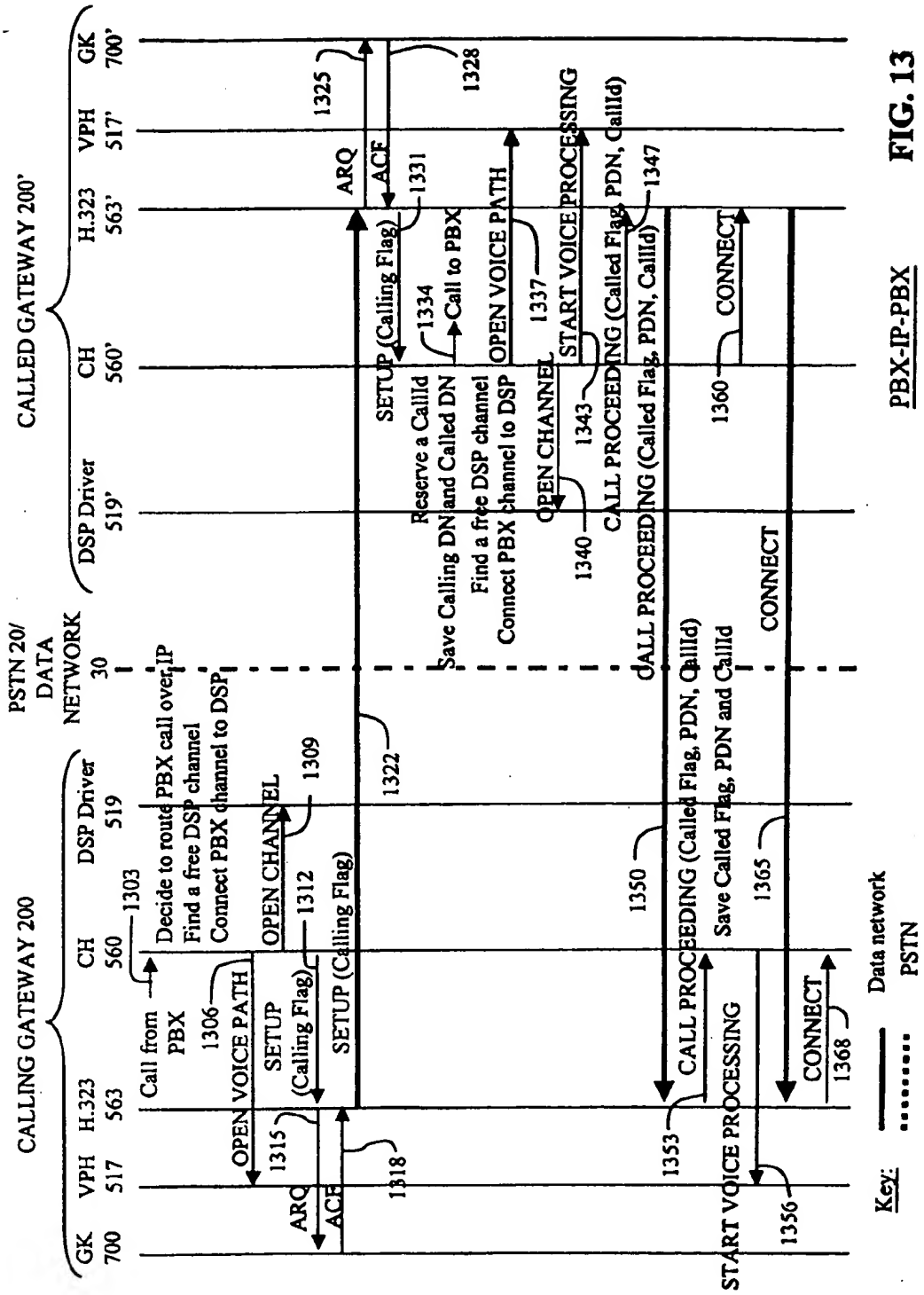

CALLING GATEWAY 200
CALLED GATEWAY 200'
PSTN 20/ DATA NETWORK 30
GK 700
VPH 517
H.323 563
CH 560
DSP Driver 519
DSP Driver 519'
CH 560'
H.323 563'
VPH 517'
GK 700'
Call from PBX
1303
Decide to route PBX call over IP
Find a free DSP channel
Connect PBX channel to DSP
1306
OPEN VOICE PATH
SETUP (Calling Flag)
OPEN CHANNEL
1312
1309
SETUP (Calling Flag)
1322
1315
ARQ
ACF
1318
Reserve a CallId
Save Calling DN and Called DN
Find a free DSP channel
Connect PBX channel to DSP
SETUP (Calling Flag)
1334
Call to PBX
1331
ARQ
ACF
1325
1328
OPEN VOICE PATH
1337
OPEN CHANNEL
START VOICE PROCESSING
1340
1343
CALL PROCEEDING (Called Flag, PDN, CallId)
1347
1350
CALL PROCEEDING (Called Flag, PDN, CallId)
Save Called Flag, PDN and CallId
1360
CONNECT
1365
CONNECT
1368
1353
START VOICE PROCESSING
1356
Key:
Data network
PSTN
PBX-IP-PBX

FIG. 13

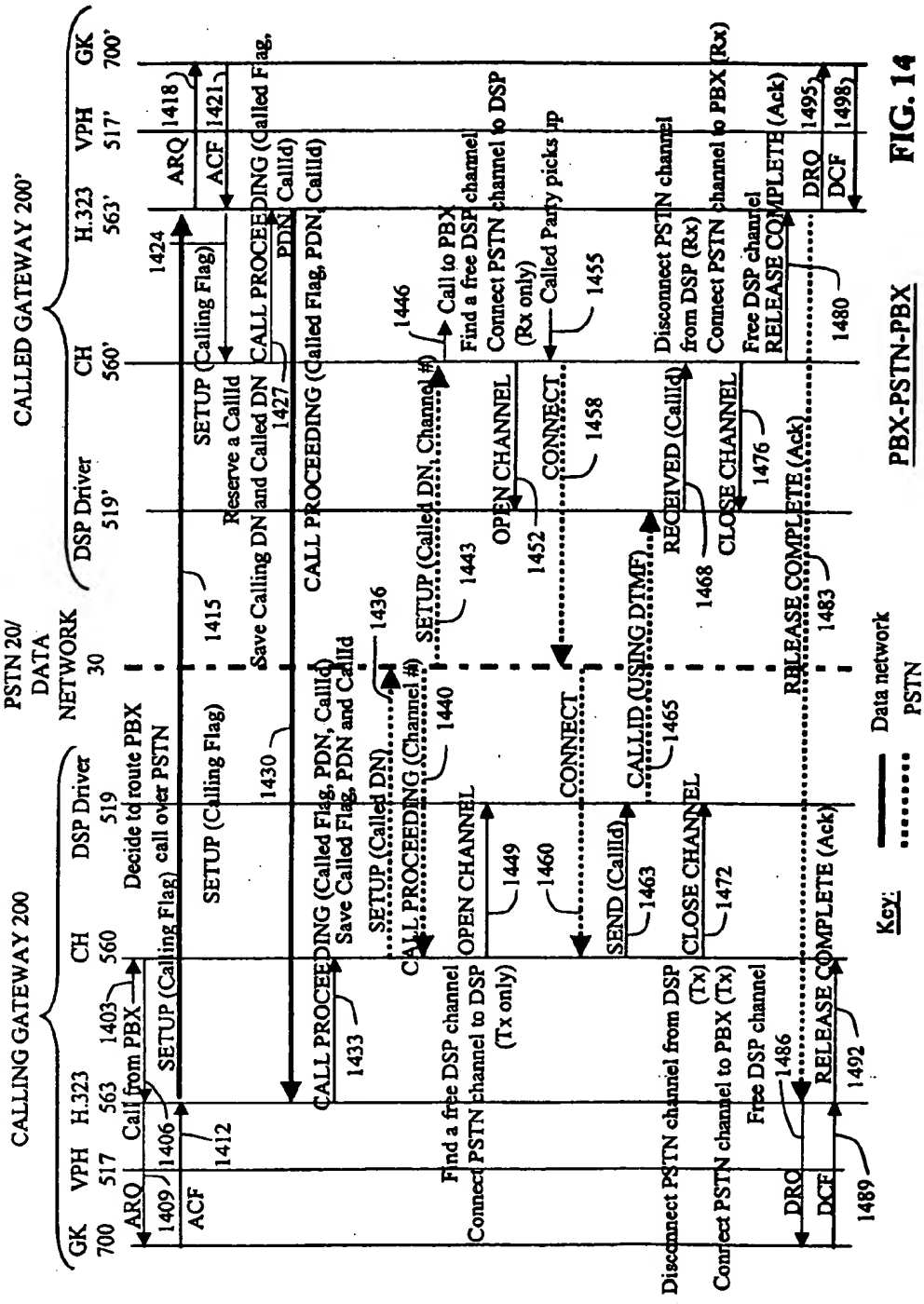

CALLING GATEWAY 200
CALLED GATEWAY 200'
PSTN 20/ DATA NETWORK 30
GK 700
VPH 517
H.323 563
CH 560
DSP Driver 519
DSP Driver 519'
CH 560'
H.323 563'
VPH 517'
GK 700'
ARQ 1409
ACF 1412
SETUP (Calling Flag) Call from PBX 1403
1406
Decide to route PBX call over PSTN
SETUP (Calling Flag) 1430
SETUP (Calling Flag) 1415
Reserve a CallId
Save Calling DN and Called DN 1427
CALL PROCEEDING (Called Flag, PDN, CallId) 1424
ARQ 1418
ACF 1421
CALL PROCEEDING (Called Flag, PDN, CallId)
Save Called Flag, PDN and CallId 1433
CALL PROCEEDING (Called Flag, PDN, CallId) 1436
SETUP (Called DN) 1440
CALL PROCEEDING (Channel #)
SETUP (Called DN, Channel #) 1443
Call to PBX Find a free DSP channel 1446
Find a free DSP channel Connect PSTN channel to DSP (Tx only)
OPEN CHANNEL 1449
OPEN CHANNEL 1452
Connect PSTN channel to DSP (Rx only)
Called Party picks up 1455
CONNECT 1460
CONNECT 1458
SEND (CallId) 1463
CALLID (USING DTMF) 1465
RECEIVED (CallId) 1468
Disconnect PSTN channel from DSP (Tx)
Disconnect PSTN channel from DSP (Rx)
Connect PSTN channel to PBX (Tx)
CLOSE CHANNEL 1472
CLOSE CHANNEL 1476
Connect PSTN channel to PBX (Rx)
Free DSP channel 1486
Free DSP channel RELEASE COMPLETE (Ack) 1480
RELEASE COMPLETE (Ack) 1483
DRQ 1495
DCF 1498
DRQ 1489
DCF 1492
RELEASE COMPLETE (Ack)
Key: Data network PSTN
PBX-PSTN-PBX

FIG. 14

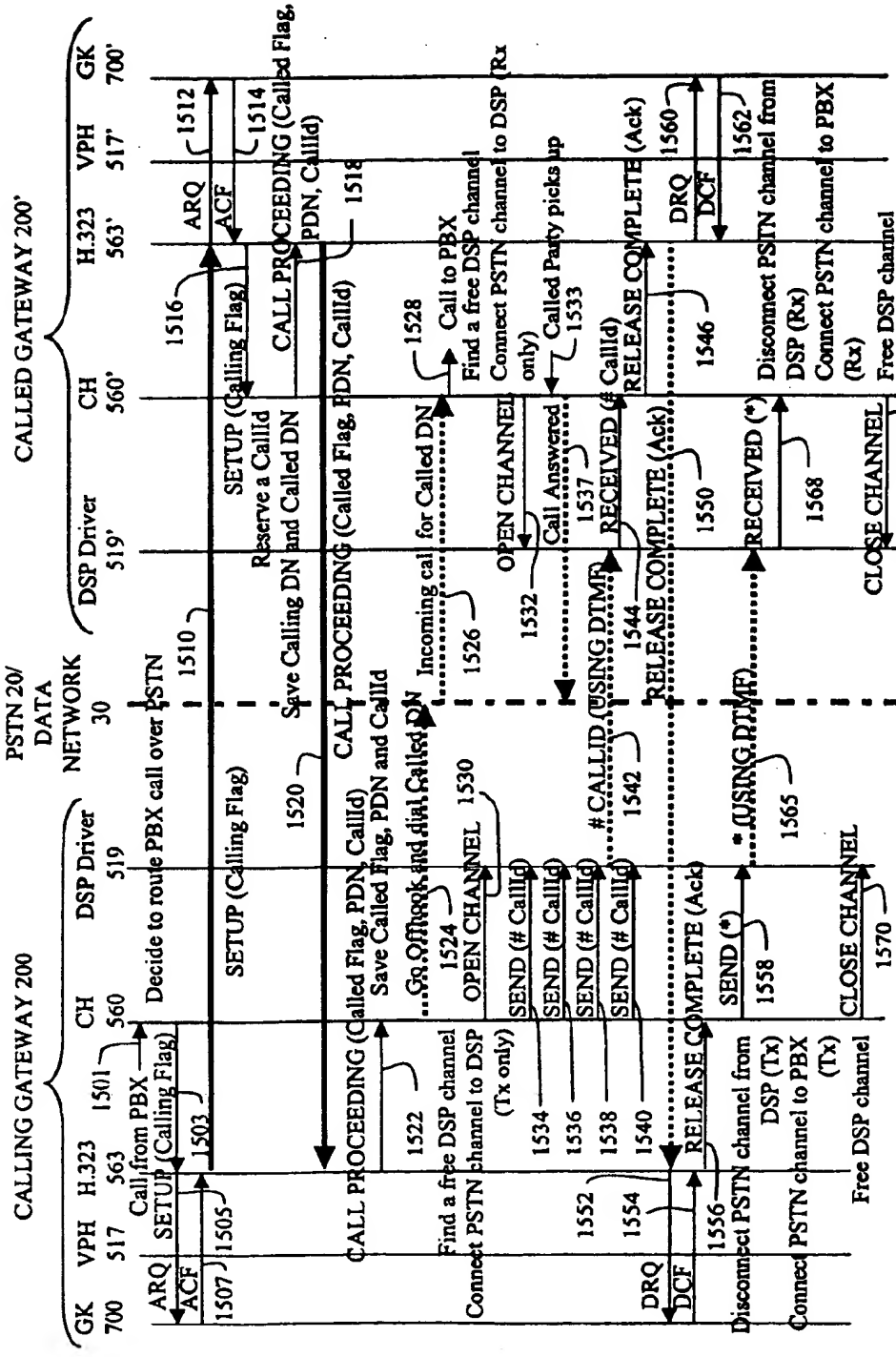

CALLING GATEWAY 200
CALLED GATEWAY 200'
PSTN 20/ DATA NETWORK 30
GK 700
VPH 517
H.323 563
CH 560
DSP Driver 519
DSP Driver 519'
CH 560'
H.323 563'
VPH 517'
GK 700'
ARQ
ACF
SETUP
Call from PBX
SETUP (Calling Flag)
Decide to route PBX call over PSTN
SETUP (Calling Flag)
Reserve a CallId
Save Calling DN and Called DN
CALL PROCEEDING (Called Flag, PDN, CallId)
CALL PROCEEDING (Called Flag, PDN, CallId)
Save Called Flag, PDN and CallId
Go Offhook and dial Called DN
Incoming call for Called DN
Call to PBX
Find a free DSP channel
Connect PSTN channel to DSP (Tx only)
Connect PSTN channel to DSP (Rx only)
Called Party picks up
Call Answered
CALLID (USING DTMF)
RECEIVED (# CallId)
RELEASE COMPLETE (Ack)
RELEASE COMPLETE (Ack)
DRQ
DCF
SEND (*)
* (USING DTMF)
RECEIVED (*)
Disconnect PSTN channel from DSP (Tx)
Disconnect PSTN channel from DSP (Rx)
Connect PSTN channel to PBX (Tx)
Connect PSTN channel to PBX (Rx)
Free DSP channel
CLOSE CHANNEL
PBX-IP-PBX
Key:
Data network
PSTN
(without connect message delivered to calling side)

FIG. 15

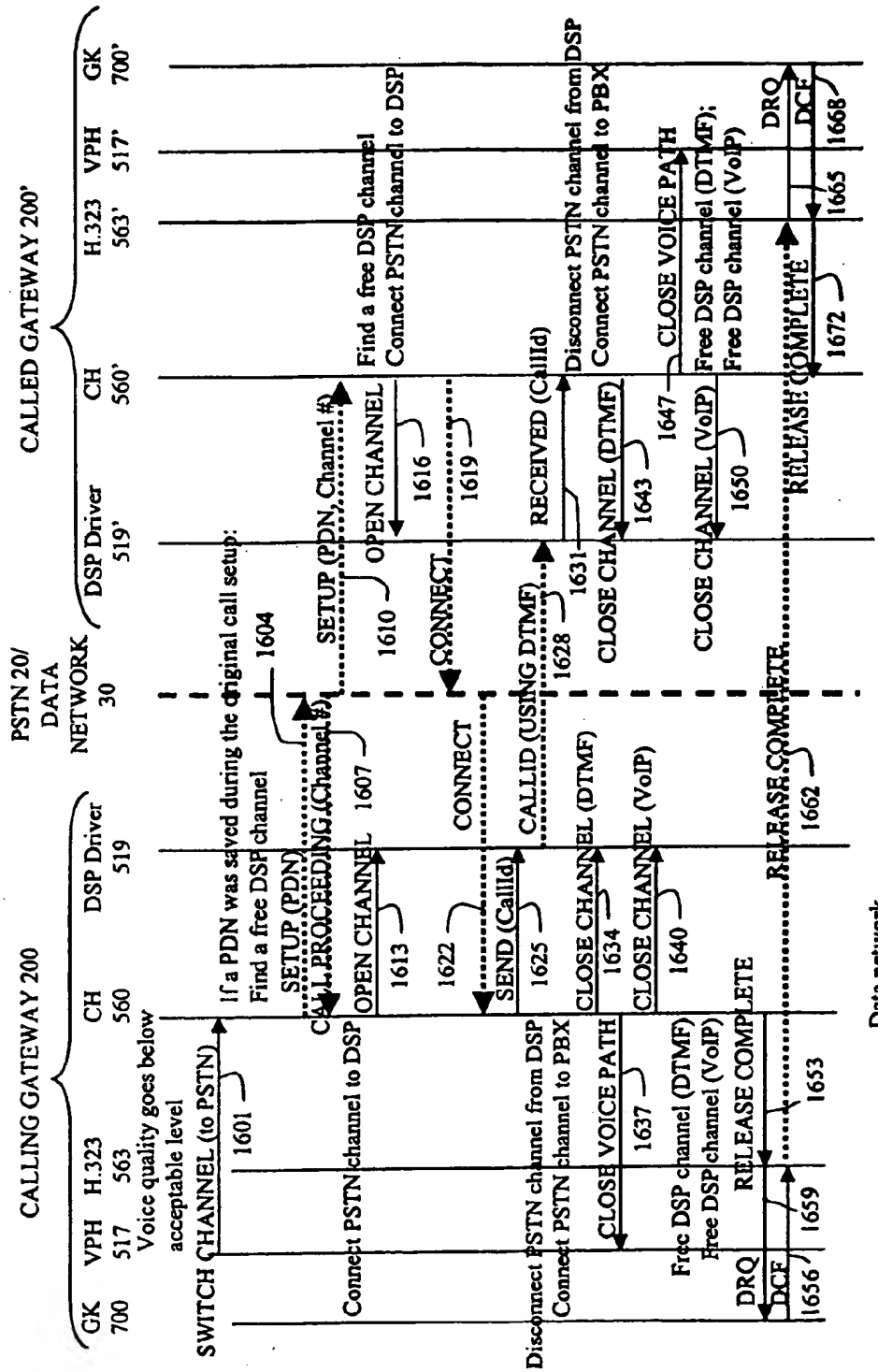

CALLING GATEWAY 200
CALLED GATEWAY 200'
PSTN 20/ DATA NETWORK 30
GK 700
VPH 517
H.323 563
CH 560
DSP Driver 519
DSP Driver 519'
CH 560'
H.323 563'
VPH 517'
GK 700'
Voice quality goes below acceptable level
SWITCH CHANNEL (to PSTN) 1601
If a PDN was saved during the original call setup:
Find a free DSP channel
1604
SETUP (PDN, Channel #)
CALL PROCEEDING (Channel #)
1607
SETUP (PDN, Channel #)
1610
Find a free DSP channel
Connect PSTN channel to DSP
OPEN CHANNEL
1613
OPEN CHANNEL
1616
Connect PSTN channel to DSP
CONNECT
1622
CONNECT
1619
SEND (CallId)
1625
CALLID (USING DTMF)
1628
RECEIVED (CallId)
1631
Disconnect PSTN channel from DSP
Connect PSTN channel to PBX
CLOSE CHANNEL (DTMF)
1634
CLOSE CHANNEL (DTMF)
1643
Disconnect PSTN channel from DSP
Connect PSTN channel to PBX
CLOSE VOICE PATH
1637
CLOSE CHANNEL (VoIP)
1640
CLOSE VOICE PATH
1647
Free DSP channel (DTMF)
Free DSP channel (VoIP)
CLOSE CHANNEL (VoIP)
1650
Free DSP channel (DTMF);
Free DSP channel (VoIP)
DRQ
DCF
1656
1659
RELEASE COMPLETE
1653
RELEASE COMPLETE
1662
RELEASE COMPLETE
1672
DRQ
DCF
1665
1668
Key:
Data network
PSTN
IP to PSTN switchover using pooled DN

FIG. 16

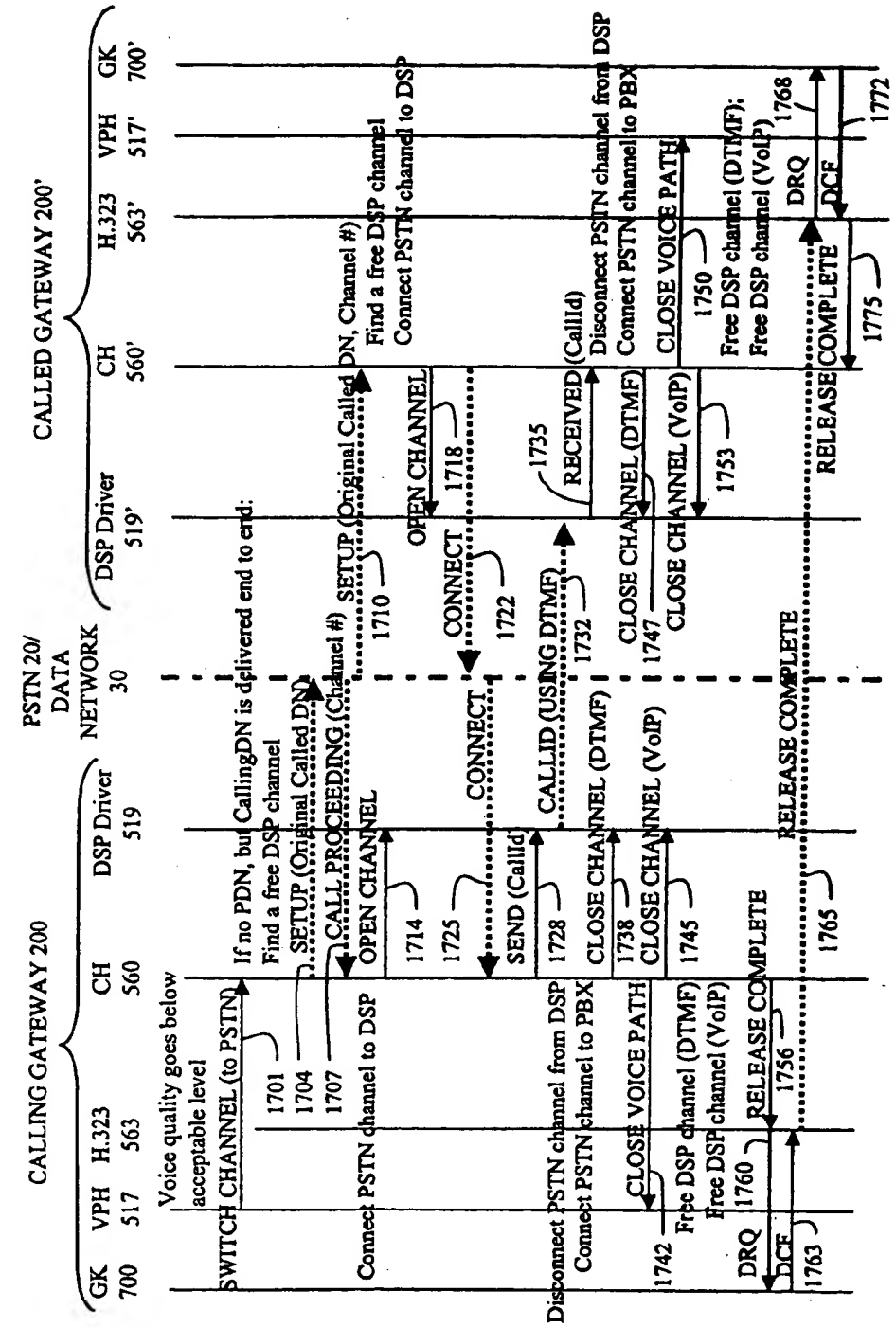

CALLING GATEWAY 200
CALLED GATEWAY 200'
PSTN 20/ DATA NETWORK 30
GK 700
VPH 517
H.323 563
CH 560
DSP Driver 519
DSP Driver 519'
CH 560'
H.323 563'
VPH 517'
GK 700'
Voice quality goes below acceptable level
SWITCH CHANNEL (to PSTN)
1701
If no PDN, but CallingDN is delivered end to end:
Find a free DSP channel
1704
SETUP (Original Called DN)
1707
CALL PROCEEDING (Channel #)
OPEN CHANNEL
1714
Connect PSTN channel to DSP
1710
SETUP (Original Called DN, Channel #)
Find a free DSP channel
Connect PSTN channel to DSP
OPEN CHANNEL
1718
1725
CONNECT
1722
CONNECT
SEND (CallId)
1728
CALLID (USING DTMF)
1732
1735
RECEIVED (CallId)
Disconnect PSTN channel from DSP
Connect PSTN channel to PBX
1738
CLOSE CHANNEL (DTMF)
1747
CLOSE CHANNEL (DTMF)
Disconnect PSTN channel from DSP
Connect PSTN channel to PBX
1742
CLOSE VOICE PATH
1745
CLOSE CHANNEL (VoIP)
1753
CLOSE VOICE PATH
1750
Free DSP channel (DTMF)
Free DSP channel (VoIP)
1760
DRQ
1756
RELEASE COMPLETE
RELEASE COMPLETE
1765
1768
DRQ
1763
DCF
DCF
1772
1775
Key:
Data network
PSTN
IP to PSTN switchover using called DN

FIG. 17

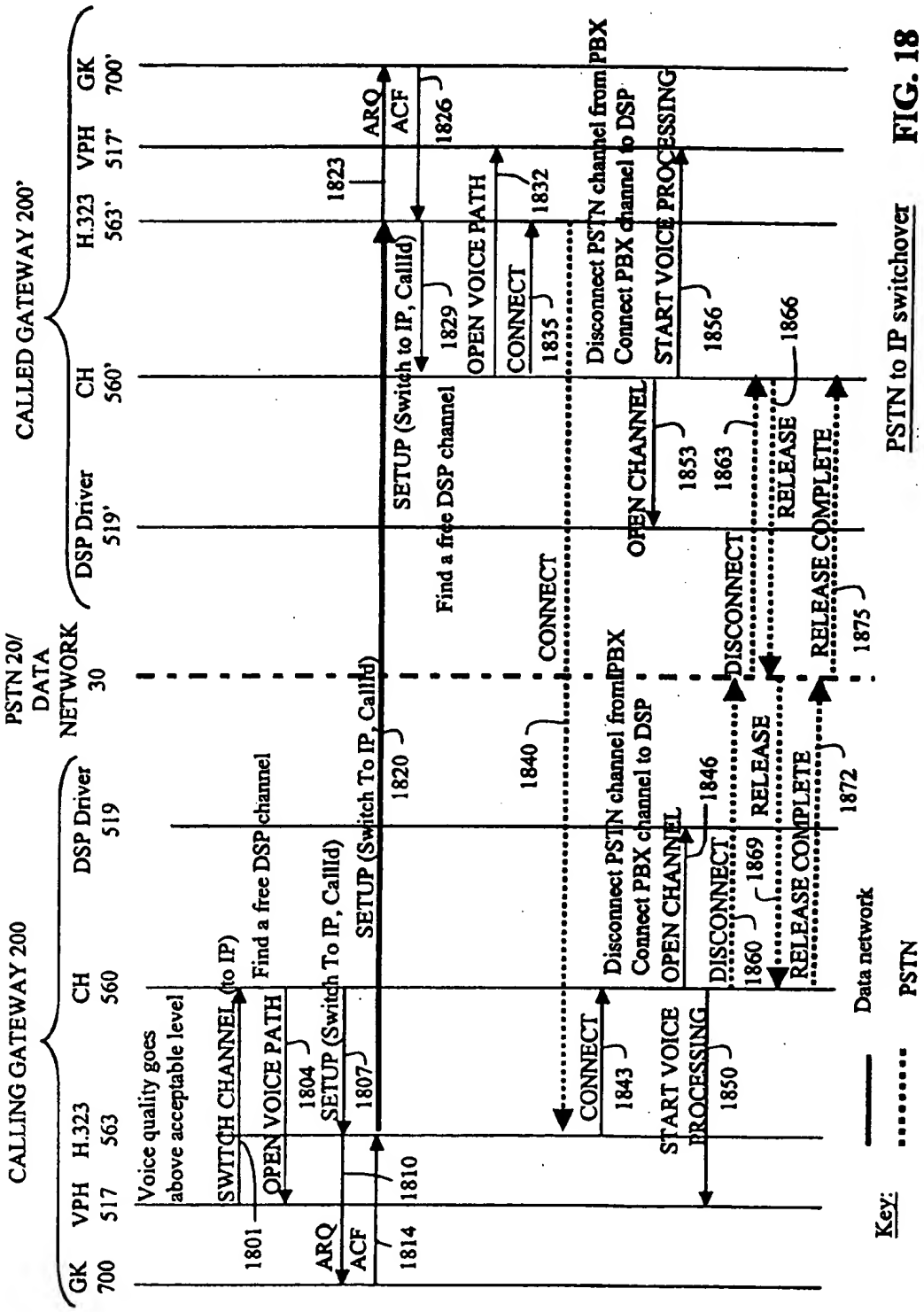

CALLING GATEWAY 200
CALLED GATEWAY 200'
PSTN 20/ DATA NETWORK 30
GK 700
VPH 517
H.323 563
CH 560
DSP Driver 519
DSP Driver 519'
CH 560'
H.323 563'
VPH 517'
GK 700'
Voice quality goes above acceptable level
SWITCH CHANNEL (to IP)
Find a free DSP channel
OPEN VOICE PATH
SETUP (Switch To IP, CallId)
ARQ
ACF
SETUP (Switch to IP, CallId)
Find a free DSP channel
OPEN VOICE PATH
CONNECT
Disconnect PSTN channel from PBX
Connect PBX channel to DSP
START VOICE PROCESSING
CONNECT
Disconnect PSTN channel from PBX
Connect PBX channel to DSP
OPEN CHANNEL
START VOICE PROCESSING
DISCONNECT
RELEASE
RELEASE COMPLETE
DISCONNECT
RELEASE
RELEASE COMPLETE
1801
1804
1807
1810
1814
1820
1823
1826
1829
1832
1835
1840
1843
1846
1850
1853
1856
1860
1863
1866
1869
1872
1875
Key:
Data network
PSTN
PSTN to IP switchover

FIG. 18

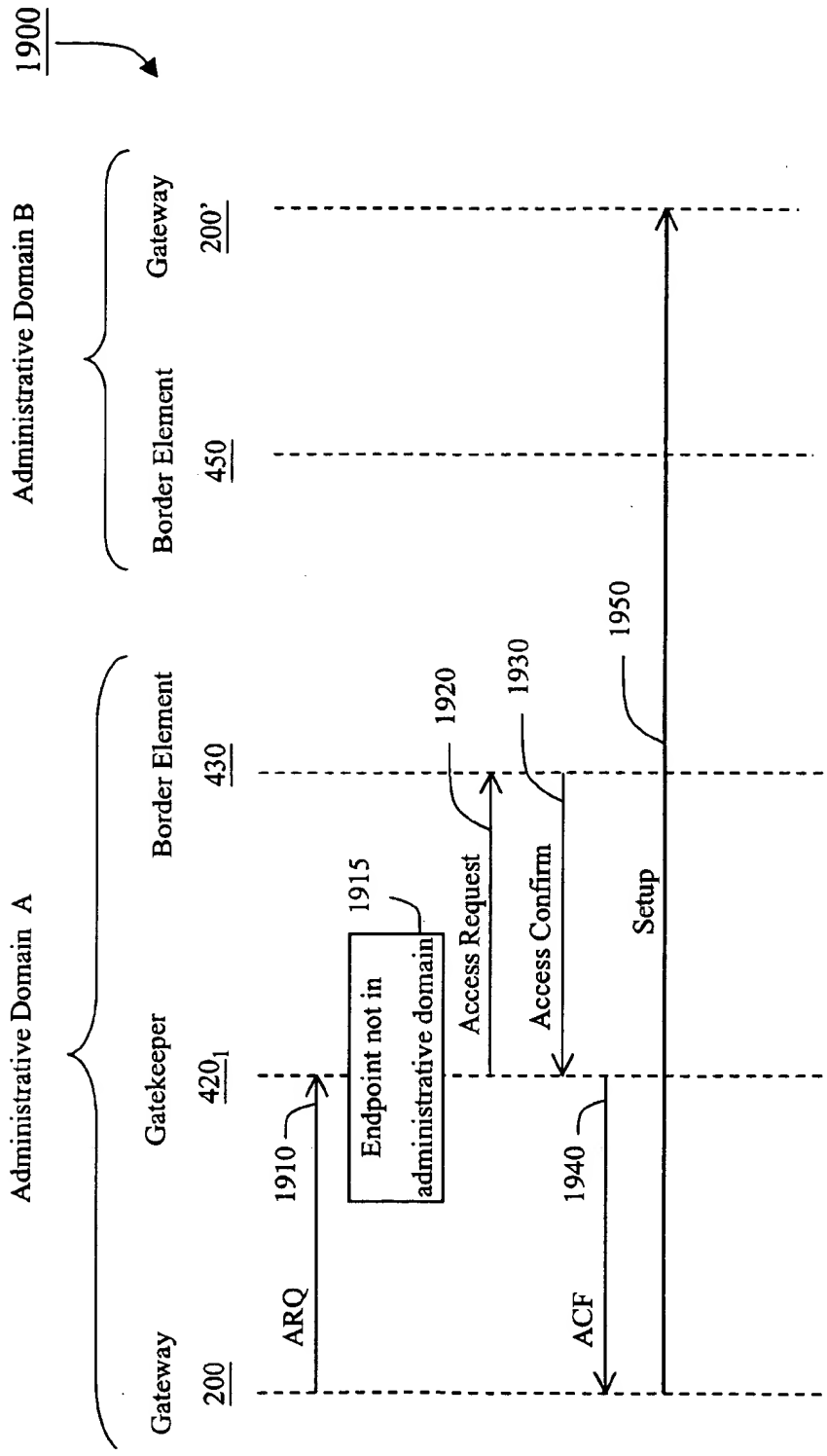


FIG. 19 Simple inter-domain call routing sequence

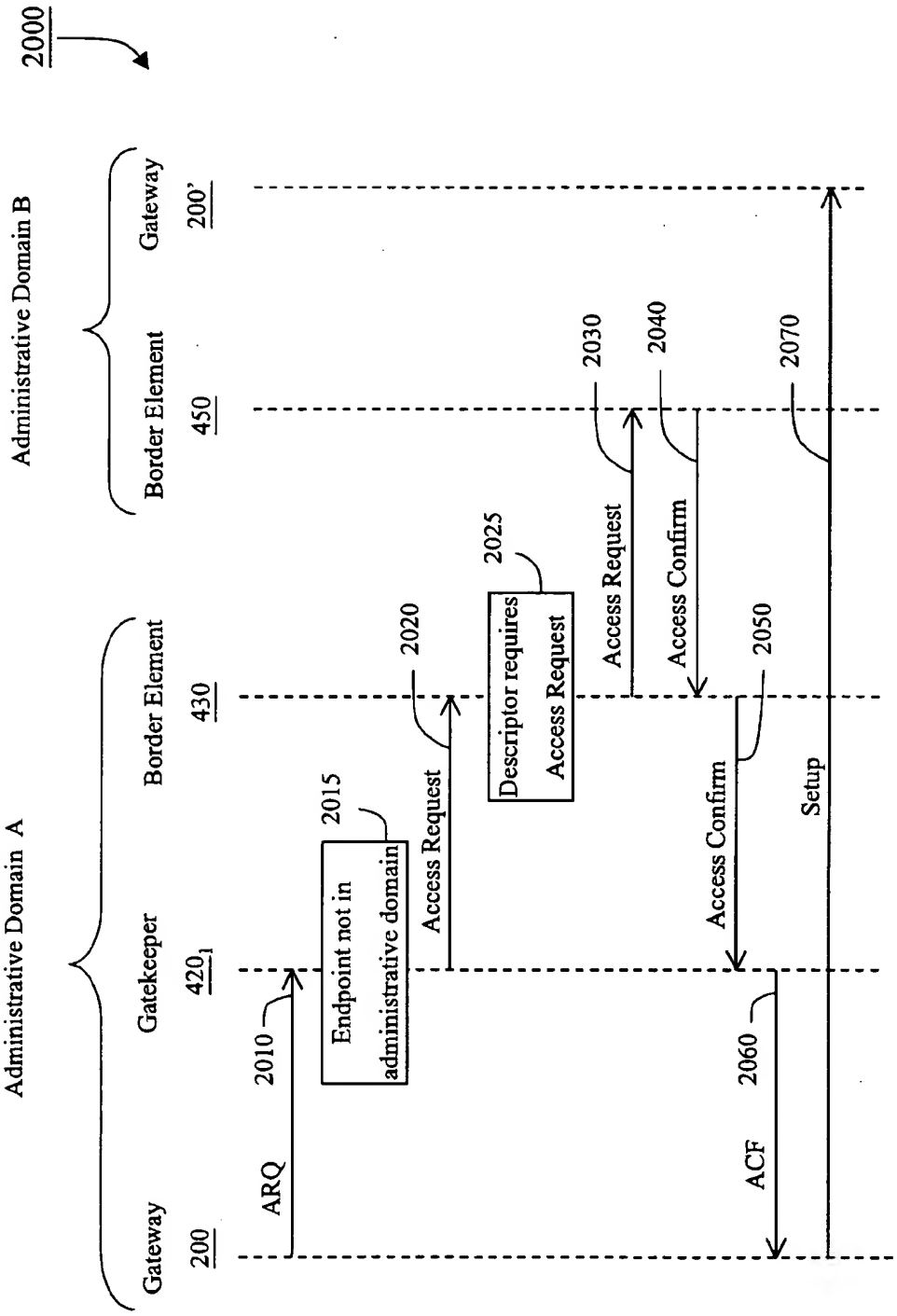


FIG. 20 Complex inter-domain call routing sequence

FIG. 21 Service Establishment sequence

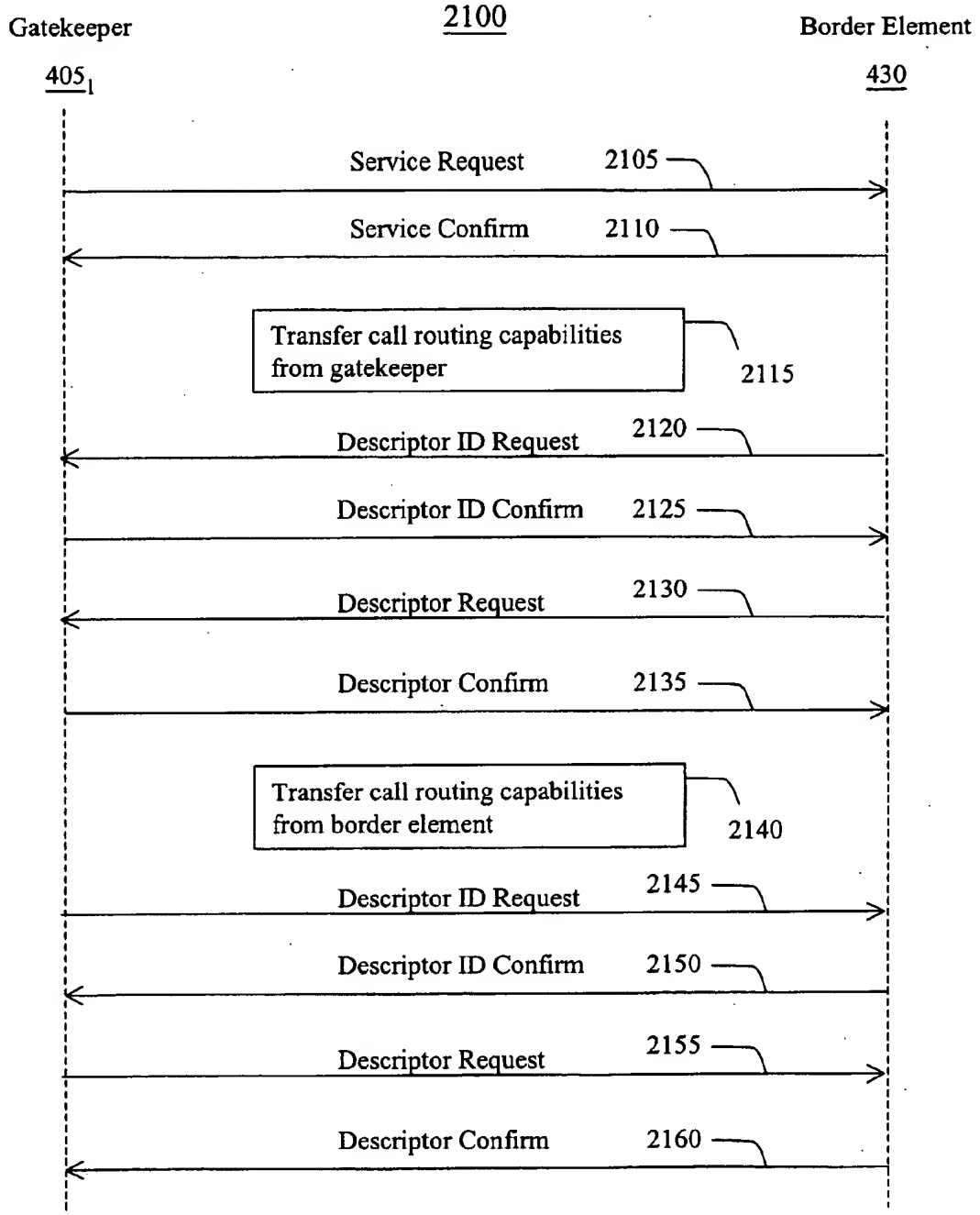


FIG. 22 Information transfer sequence

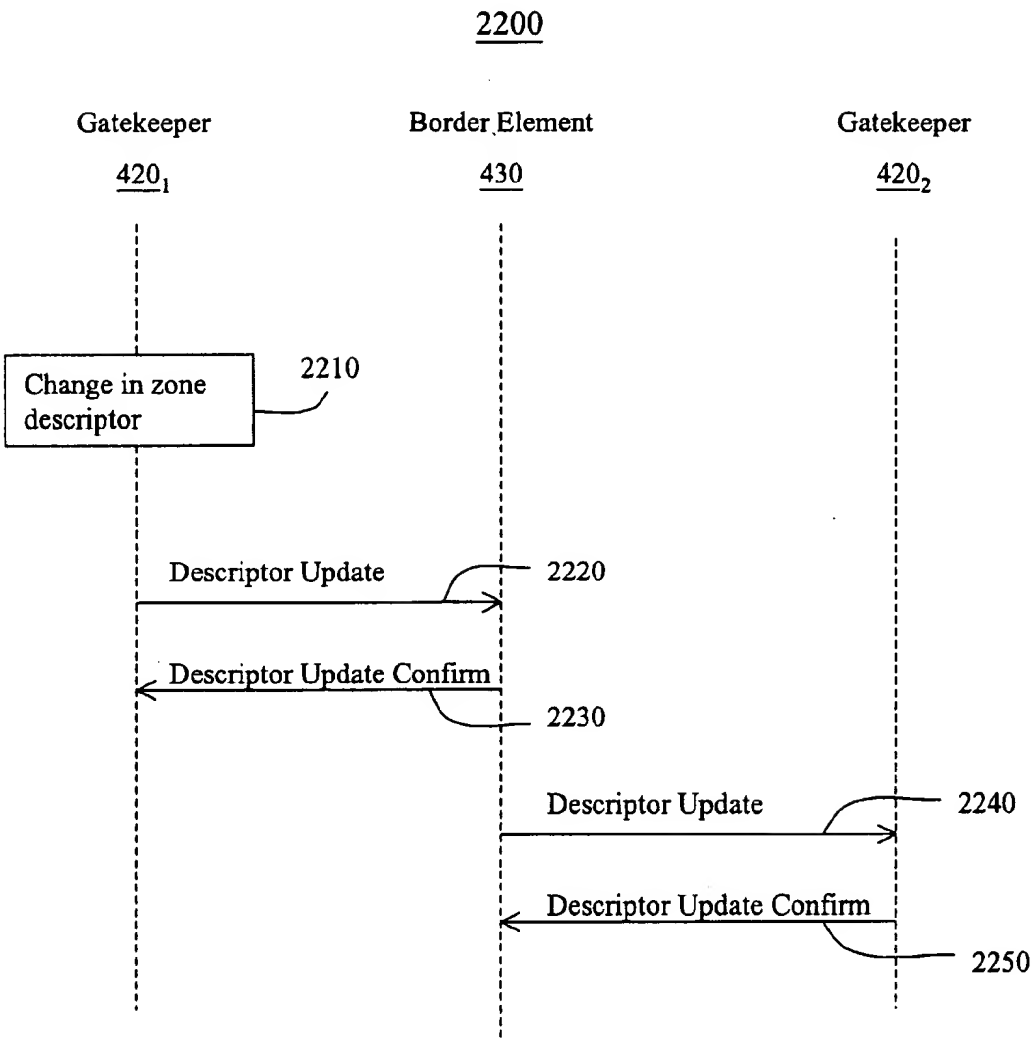
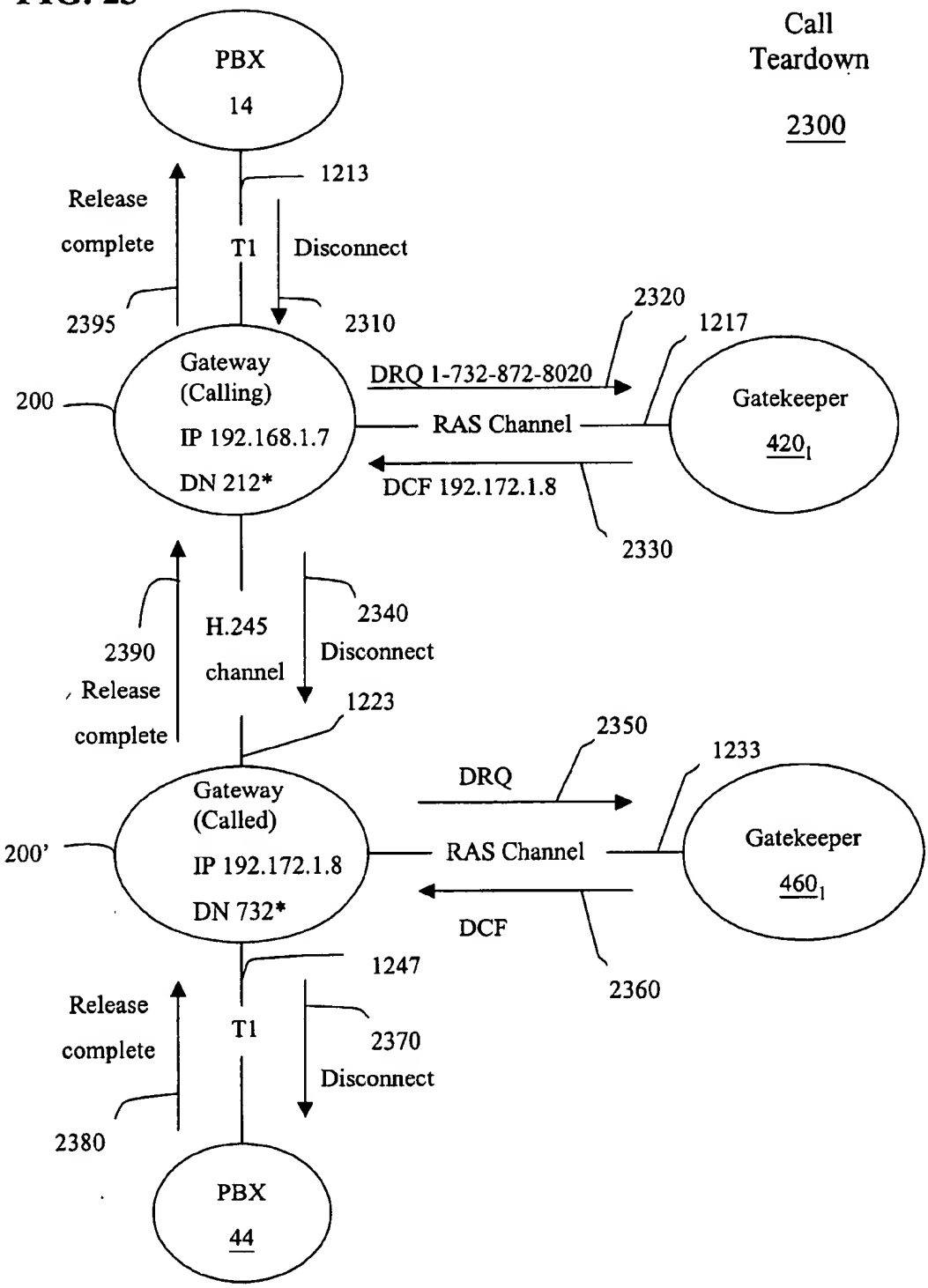
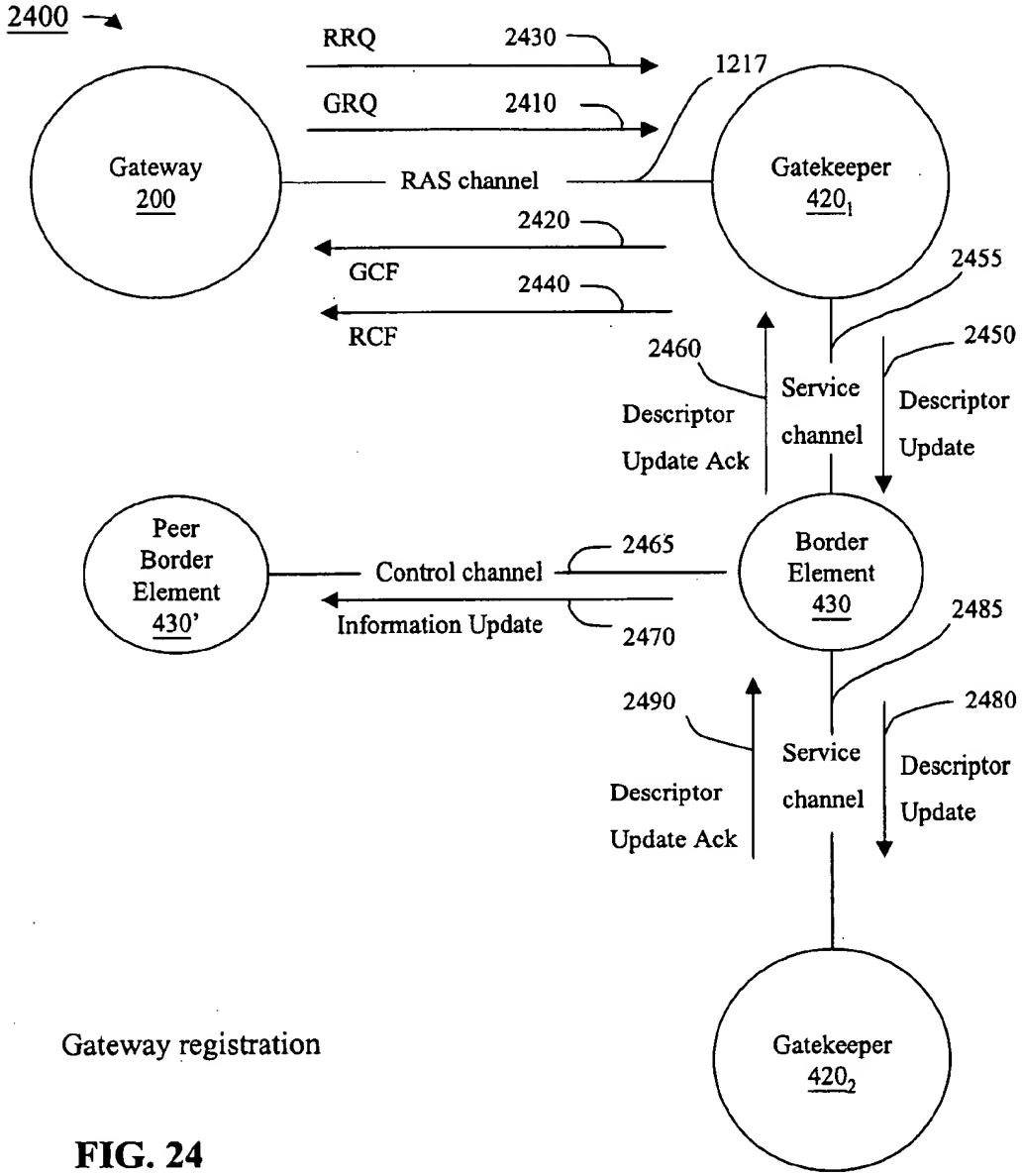


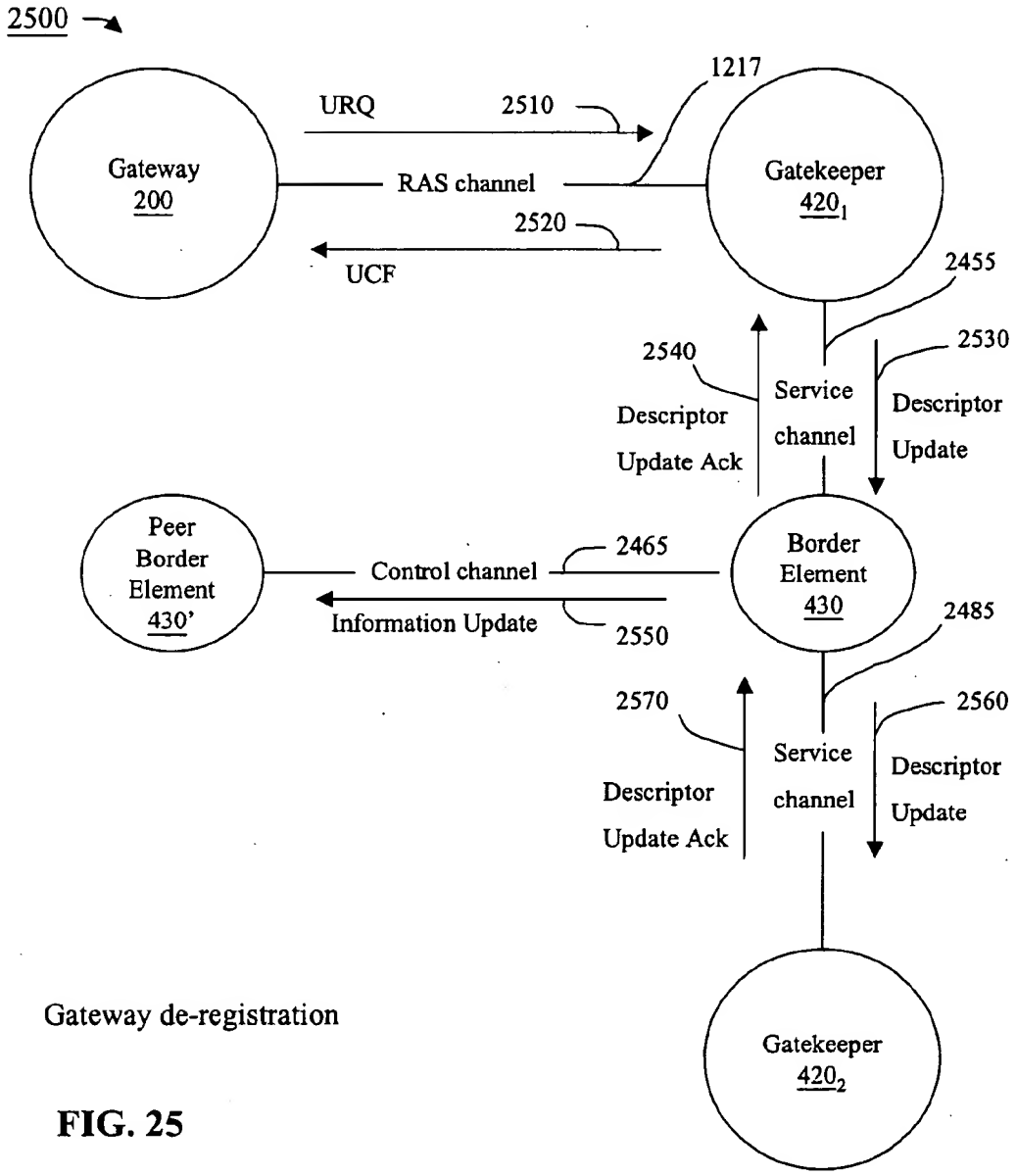
FIG. 23


Call
Teardown
2300
PBX
14
1213
Release
complete
T1
Disconnect
2395
2310
2320
1217
Gateway
(Calling)
IP 192.168.1.7
DN 212*
200
DRQ 1-732-872-8020
RAS Channel
DCF 192.172.1.8
Gatekeeper
420$_1$
2330
2390
Release
complete
H.245
channel
2340
Disconnect
1223
2350
1233
DRQ
Gateway
(Called)
IP 192.172.1.8
DN 732*
200'
RAS Channel
DCF
Gatekeeper
460$_1$
2360
1247
Release
complete
T1
2370
Disconnect
2380
PBX
44



Gateway registration

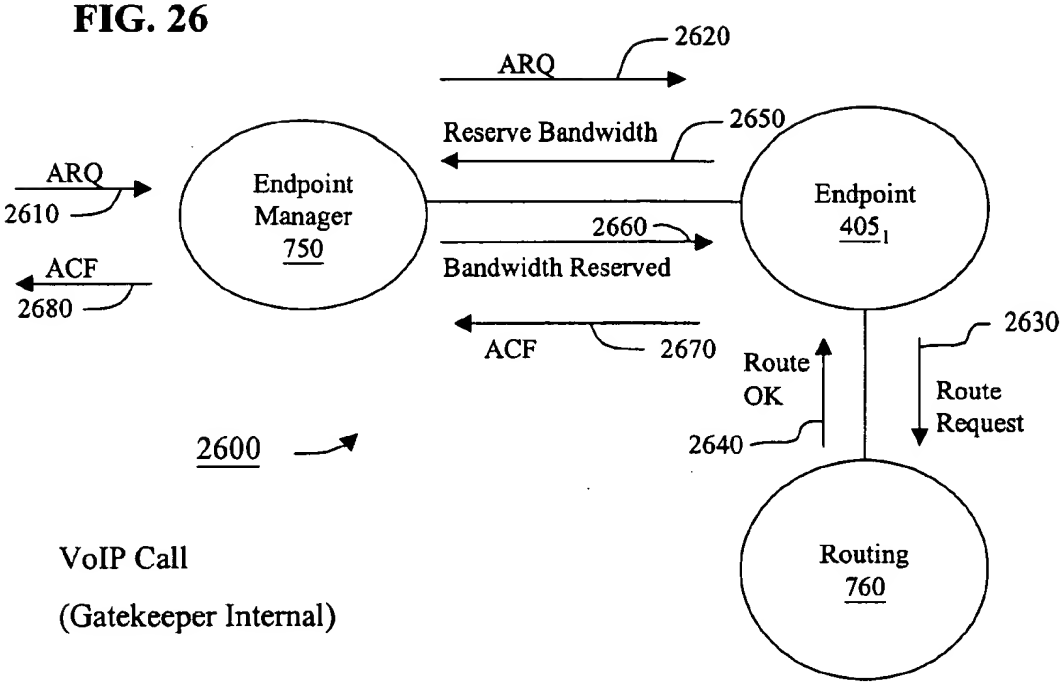
FIG. 24



Gateway de-registration

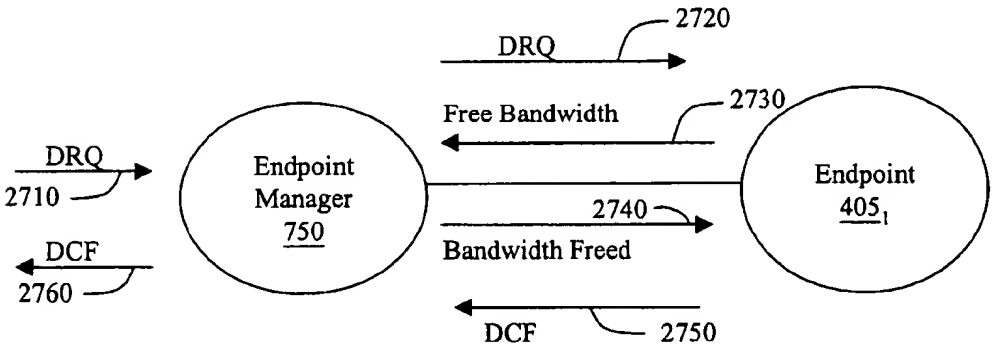
FIG. 25

FIG. 26



VoIP Call

(Gatekeeper Internal)



2700

Call Teardown

(Gatekeeper Internal)

FIG. 27

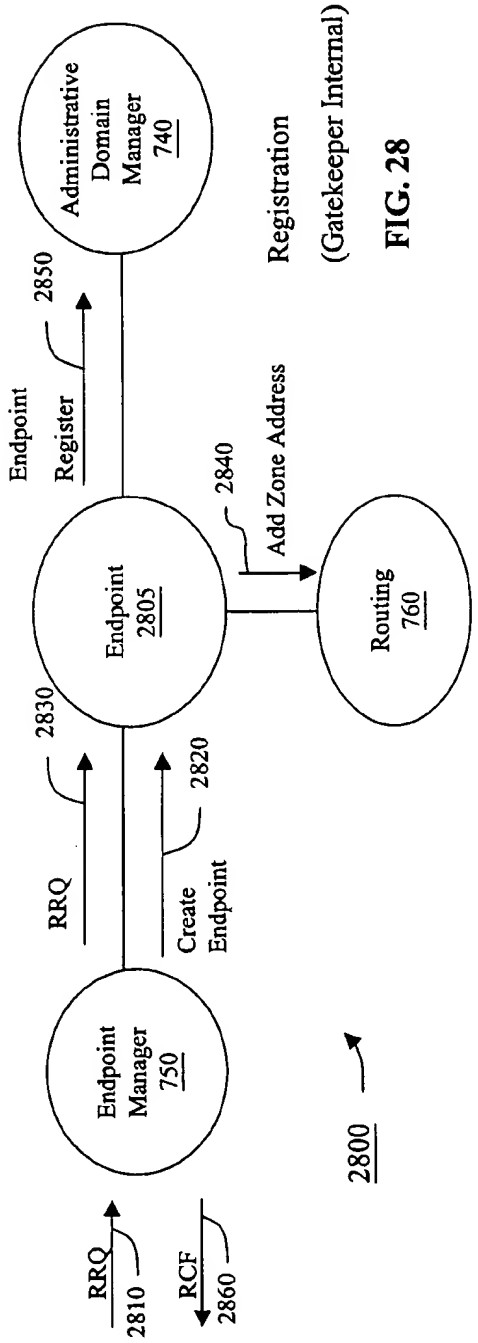

Registration
(Gatekeeper Internal)
Administrative Domain Manager 740
Endpoint 2805
Routing 760
Endpoint Manager 750
Endpoint Register 2850
Add Zone Address 2840
RRQ 2830
Create Endpoint 2820
RRQ 2810
RCF 2860
2800

FIG. 28

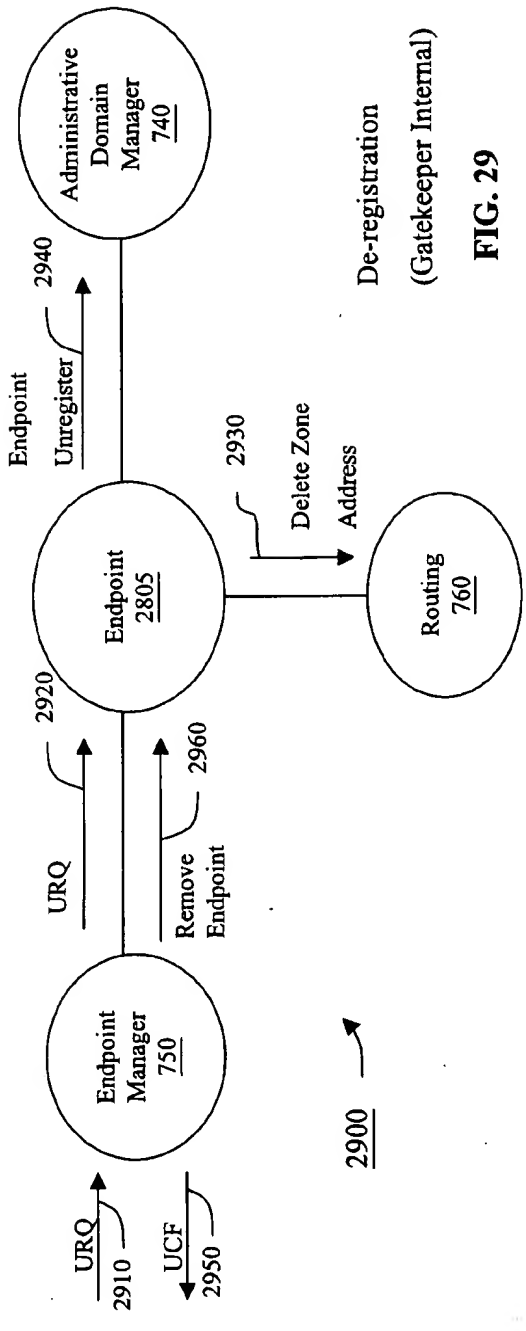

De-registration
(Gatekeeper Internal)
Administrative Domain Manager 740
Endpoint 2805
Routing 760
Endpoint Manager 750
Endpoint Unregister 2940
Delete Zone Address 2930
URQ 2920
Remove Endpoint 2960
URQ 2910
UCF 2950
2900

FIG. 29

OKAPPARATUS FOR A VOICE OVER IP (VOIP) TELEPHONY GATEWAY AND METHODS FOR USE THEREIN

BACKGROUND OF THE DISCLOSURE

1. Field of the Invention

The invention relates to apparatus, and accompanying methods for use therein, for a telephony gateway intended for use, e.g., paired use, at opposite ends of a data network connection, in conjunction with at each end, e.g., a private branch exchange (PBX) for automatically routing telephone calls, e.g., voice, data and facsimile, between two peer PBXs over either a public switched telephone network (PSTN) or a data network, based on, among other aspects, cost considerations for handling each such call and called directory numbers, monitoring quality of service (QoS) then provided through the data network and switching ("auto-switching") such calls back and forth between the PSTN and the data network, as needed, in response to dynamic changes in the QoS such that the call is carried over a connection then providing a sufficient QoS.

2. Description of the Prior Art

Over the past century, telephone communications have become rather ubiquitous as the public switched telephone network (PSTN) has expanded into increasingly rural and other remote areas of the country, thus affording nearly universal telephone access. The PSTN provides real-time circuit-switched connections between caller and called parties, i.e., it establishes a continuous real-time link between caller and called locations, the latter often being specified by a string of digits entered by the caller; maintains that connection for the duration of a telephone call and then tears down that connection once that call terminates.

While basic plain old telephone service (POTs) connections typically provide continuous high quality analog connections, suited for voice, facsimile and relatively low speed data, such connections, based on their toll charges, can be expensive to use. Telephone companies frequently price these connections based on distance and time, i.e., a distance between the caller and called locations and duration of each call. Over the past few years in the United States, competition among regional and long distance telephone companies has existed and is intensifying, so much so as to effectively, in many instances, reduce telephone toll charges. However, such competition is only now emerging in many foreign countries. Further, various foreign governments have set relatively high interconnection tariffs to protect their local telephone companies, which are frequently governmentally regulated monopolies, from competitive pricing pressures arising from foreign carriers. Consequently, while telephone charges, on a per minute basis, are relatively inexpensive in the United States, the same is not true for telephone calls within and between foreign countries. In that regard, international calls between one country and another, such as the United States, can be rather expensive.

For many types of communication, such as data, continuous real-time switched connections, provided by the PSTN, are simply not necessary, given, e.g., relaxed latency restrictions for data, and are too costly.

Hence, within the last decade, private packet networks (commonly referred to as private "data" networks) have experienced phenomenal growth as organizations, particularly those with computer and other digital equipment stationed at disparately located offices, sought cost-effective methods of communicating digital information between these offices. For ease of use and to accommodate as wide

a universe of currently available network equipment and computer software as possible, these networks are generally designed to embody Internet Protocol (IP) based routing (which is the same methodology used in the Internet).

Though initial costs associated with implementing a private data network can be significant, average per use charges incurred through use of such a network tend to be considerably less than the toll charges for similar carriage, in terms of an amount of information being communicated, associated with the PSTN and hence, if the private network is sufficiently well used, can provide substantial cost savings to its owner as compared with equivalent use of the PSTN.

During the course of designing a private data network, various long-haul communication links that underlie the network are often chosen to provide bandwidth which, to accommodate anticipated growth, greatly exceeds current usage requirements. A common result of this is that many organizations, which have private data networks in operation, find themselves with significant amounts of unused (excess) installed bandwidth, which they have already built into their cost structure, available on their networks. Hence, some amount of additional traffic can be carried over this available bandwidth at what is, for all intents and purposes, essentially no additional cost. However, bandwidth is ephemeral: it is either consumed or not; it can not be stored for future use and hence, if not used when it is available, is simply wasted.

Those organizations that have implemented and use private data networks also tend to be extremely heavy telephone users as well, thus incurring substantial telephone charges on a regular continuing basis. These organizations include relatively large corporations, as well as government, academic and military organizations. Moreover, with increasing global computerization caused by explosive proliferation of personal computer usage over the past decade, even mid-sized and relatively small organizations with multiple offices are increasingly experiencing a need for access to an IP-based data network to facilitate inter-office data sharing and data communication. Though these organizations rarely, if ever, have sufficient usage to justify implementing their own networks, they are increasingly turning, for reasons of security and economy, to various network providers who offer secure access to a shared private IP network.

Recognizing the substantial telephone charges which these organizations regularly incur, particularly when viewed in the context of excess bandwidth available on their private data networks (whether dedicated or shared) and a near zero marginal cost of utilizing that bandwidth, these organizations would likely stand to economically benefit if this bandwidth could be used in some fashion to carry telephone calls that would otherwise be routed, at much higher cost particularly for international traffic, through the PSTN.

Currently, an effort, commonly referred to as "Voice over IP" (or more simply just "VoIP"), is underway in the art to develop technology and ultimately commercial products that can be utilized to transport, as an alternative to use of the PSTN, voice, data and facsimile communication, which would heretofore be carried over the PSTN, in packetized fashion over an IP data network, such as the Internet or a private data network. As currently envisioned in the art and described in A. Cary, "IP PBXs: Open Questions", *Data Communications*, March 1999, pages 69–83 and particularly page 72, products embodying this technology will probably utilize one of two basic approaches: (a) an "adjunct"

approach, and (b) a LAN-based approach. The adjunct approach would use existing subscriber PBXs, subscriber line wiring and telephone sets but incorporate a VoIP telephony gateway, as an "adjunct", at each of a number different sites. At each site, a corresponding gateway would be situated between PSTN trunk connections to a PBX at that site and connections to an IP network, so as to route incoming and outgoing telephone calls between PBX peers at these sites through the IP network. In contrast, the LAN-based approach would replace conventional telephone subscriber equipment and telephone PBXs with IP-compatible telephones to packetize voice calls, and carry these calls over local area networks (LANs).

The LAN based approach is likely to meet with significant disfavor and commercial skepticism owing to a substantial expense, particularly with large organizations that have extensive telephone systems, associated with removing and replacing existing telephone equipment, including PBXs and telephone instruments. This will be particularly true if, as we believe, the end-user price of a VoIP telephony gateway can kept to a reasonable level. Should this occur, the adjunct approach, by requiring a significantly reduced capital outlay while potentially providing substantial savings on telephone toll charges, will likely be widely adopted in the market and hence experience significant, widespread and rapid commercial success.

While carriage of telephony traffic over an IP network clearly holds theoretical promise and economic attraction, particularly through use of an "adjunct" approach, several obstacles exist, of which the following are illustrative. Any of these obstacles, if not properly addressed, could seriously hamper practical implementation and eventual deployment of this approach.

First, quality of service associated with a data connection provided through an IP network can vary widely. Such a connection can experience wide dynamic changes in latency, jitter and/or packet loss. Given the error correction processing that usually occurs at each end of a data connection, packet traffic can usually withstand transient changes, to a fairly significant degree, caused by any of these affects, before integrity of its payload data becomes jeopardized. However, voice traffic is particular sensitive to these affects. Specifically, if packetized speech were to be subjected to transient changes in any of these affects, then this speech, once converted into an analog signal, may well contain audible distortion that might be highly objectionable to an individual on either end of a call. Consequently, any equipment that routes telephony traffic, originally destined to a PSTN, over an IP network instead must incorporate some mechanism to measure quality of service (QoS) of a networked connection, provided through the IP network, which carries telephony traffic and then switch this traffic over to the PSTN whenever the QoS of this connection sufficiently degrades. Preferably, this switchover itself should occur when distortion caused by a degradation in the QoS would likely become objectionable to a listener at either end of a call. In addition, this equipment should implement the switchover itself in a manner that is substantially inaudible, i.e., transparent, or at least not objectionable to that listener. In that regard, one illustrative device, referred to as "Selsius-IP PBX" gateway and recently developed by Cisco Systems, apparently switches a telephone call from the IP network to the PSTN should the latency on the IP network rise too far.

Second, not every telephone call needs to be routed over an IP network. In fact, no economic benefit results from routing certain calls over this network; these calls would best be handled through the PSTN. These calls include those which are, e.g., strictly local in nature, including, e.g., "911" calls, as well as those to toll-free numbers.

Furthermore, any VOIP gateway that is intended to carry telephone traffic must exhibit a very high degree of reliability and fault-tolerance, preferably similar to that of the PSTN itself.

Though efforts are currently underway at various organizations to develop a VoIP telephony gateway, between a PSTN and an IP network, to date, no commercial products appear to exist in the marketplace that implement IP telephony in a manner that remedies the above-noted obstacles.

In that regard, various gateways that have been announced seem deficient with respect to overcoming one or more of these obstacles. In that regard, one such illustrative device referred to as "NetPhone IPBX" gateway developed by NetPhone, Inc. of Marlborough, Massachusetts appears to provide a fallback capability to switch a telephone call to the PSTN from the IP network only in the event either the IP connection fails or a computer operating system, on which a software portion of the gateway executes, fails but not if QoS of the IP connection simply degrades.

No VOIP telephony gateway of which we are aware appears to be capable of selective call placement, i.e., deciding whether, from a nature of the telephone call itself, i.e., a called directory number, that call is best routed over the IP network or the PSTN and then routing the call accordingly, and/or is sufficiently reliable and fault-tolerant.

Therefore, a significant need currently exists in the art for a VOIP telephony gateway, particularly in view of the widespread adoption and substantial cost savings that could well accrue from its use, that is not only able to route a telephone call to an IP network in lieu of the PSTN but also can switch the call between these networks, as needed, based on QoS then being provided by the IP network. Such a gateway should also provide selective call placement such that those telephone calls that are not able to generate a cost savings, or other benefit, from being handled through the IP network are identified and routed to the PSTN rather than to the IP network. In addition, such a gateway should be highly reliable and fault-tolerant.

SUMMARY OF THE INVENTION

The present invention advantageously satisfies these needs, while overcoming known obstacles in the art, by providing a telephony gateway which, when operated with a similar peer gateway and each connected at an opposite ends of PSTN and data network connections, dynamically switches a call alternately between the data network and the PSTN based on real-time measurements of quality of service (QoS) then associated with the data network so as to carry the call over the particular network then providing sufficient QoS.

In accordance with our inventive teachings, once a telephone call has been initially routed to either the PSTN or to the data network (e.g., an IP network), then, should the QoS of a connection through the data network change, the call will be automatically switched ("auto-switched") to and routed through the other network, with the switching dynamically changing, during the duration of the call and in a substantially transparent manner to both the calling and called parties, alternating between the data network and the PSTN, as necessary, in response to dynamic changes in the QoS of the data network.

In particular, the inventive gateway determines network quality through dynamic measurements of latency, packet loss and error rate (jitter). Should either gateway involved in

a call determine that network quality has either increased or decreased to necessitate an auto-switch either to the data network from the PSTN or the opposite, that gateway (hereinafter, for simplicity of reference, the "calling gateway") will initiate an information exchange, using our inventive extensions to the H.323 protocol with its peer gateway (hereinafter, the "called" gateway).

Specifically, if the call is to transition from the data network to the PSTN, the called gateway will select an available directory number from a pool of directory numbers (PDN) that has been assigned to it during its configuration and convey that specific number to the calling gateway. Once the calling gateway receives the particular PDN, it originates a circuit-switched call over its PSTN trunk connection to that PDN. The called gateway, sensing an incoming call on its PDN, will determine whether this number corresponds to the particular PDN on which that gateway is now expecting a call. If it is a different PDN number from that which it is expected, that gateway sends a message to the calling gateway over the network connection and waits for a gateway to claim this call. If this call is on the correct PDN, then the called gateway switches the call from its network connection to the now established circuit-switched connection through the PSTN. Once this occurs, the data network connection for this call is torn down by both gateways as if the call were completed. Auto-switching also occurs in reverse, from the PSTN back to the data network, when network quality sufficiently improves.

As per our inventive teachings, peered gateways facilitate auto-switching of telephone calls between the PSTN and the data network by establishing call-specific information for each call, including a unique call identifier (CallId), and Calling and Called Flags and communicating that information between themselves during call setup. Gateways communicate this information by embedding this information into various H.323 messages, specifically in a so-called "nonstandard Data" field, using call independent signaling. By virtue of this information, the gateways on calling and called sides form the same association for each call routed therebetween and with a common CallId used for that call. This identifier distinguishes that call from any other then being handled by either gateway such that these two peered gateways, acting in unison, can switch this particular call between these networks, as needed, without affecting any other calls.

Specifically, through use of call independent signaling features of an H.323 standard, a Calling Flag is embedded within an H.323 SETUP message, and a Called Flag, a CallId and a selected PDN are all embedded within an H.323 CALL PROCEEDING or H.323 CONNECT message. In that regard, the contents of the Calling Flag, which are generated by a calling side, contains information, for a given call being established, which indicates, to a called side, whether, from the calling gateway, that call can be auto-switched. In response to this SETUP message, the called side generates and saves a CallId number which uniquely identifies that call and then passes that ID back to the calling side, along with the Called Flag and PDN. The Called Flag specifies whether, from the called gateway, this call can be auto-switched. The calling side then saves this information for later use in properly auto-switching the call between the data network and the PSTN, should a need to auto-switch then occur.

Our inventive gateway functions as an entity within an H.323 environment. The gateway implements at least one gatekeeper, to which the gateway registers itself, and at least one border element. The gatekeeper manages a group of endpoints which collectively constitute a zone. An administrative domain is formed of at least one gatekeeper and a border element connected to the gatekeeper(s) in the domain. The border element provides external network access into the administrative domain.

Advantageously, as a feature of our invention, for increased local redundancy, our inventive gateway also implements peered border elements. Peered border elements function together and behave as a single monolithic border element, i.e., one "logical" border element, but with their functionality being duplicated across these such elements. Hence, if either of the peered border elements in an administrative domain fails, the other peered element can provide inter-domain routing and inter-zone routing within that domain. Peered border elements preferably have a loosely coupled distributed architecture, with no hierarchical differences. All transactions from gatekeepers or one border element in a domain are shared with its peer border element. As such, transaction data stored in one peered border element remains synchronized with that stored in the other, such that either one border can immediately undertake transaction processing should its peer border element fail or be taken out of service.

Each peered border element has both TCP/IP server and client connections. Messages between peer border elements include information download and information update messages, as well as messages to establish and disconnect TCP/IP connections therebetween. The information download message is sent by one "originating" border element to its peer, upon establishing a TCP connection with that peer. This message shares all the call routing capabilities of the originating border element with its peer. The message contains local service relationships (internal to a domain), local descriptors, external service relationships (external to a domain) and external descriptors. The local service relationships define transport addresses of each of the gatekeepers that have a service relationship with the originating border element. The local descriptors define routing descriptors and are obtained from either a static configuration of the same domain as the originating border element or from gatekeepers, located within this domain, that have established service relationships with the originating border element. The external service relationships define, for the originating border element, transport addresses of those border elements external to this domain that have established a service relationship with the originating border element. The external descriptors define routing descriptors, that are obtained from either a static configuration of the H.323 environment or from border elements, located external to the domain that contains the originating border element, that have established service relationships with the originating border element. An information update message is sent from the originating border element to its peer in order to notify the latter of a change either in information affecting a gatekeeper located within the same domain or information received from a border element located external to this domain. The particular border element within an pair of "peered" border elements that originally received such information is responsible to send that information to all its peers.

Furthermore, each gateway advantageously provides, as another feature of our invention, selective call routing to route, based on called directory numbers, only those of its outgoing calls to the data network that can provide effective cost savings to the calling parties and/or their organizations. This routing is based on called number information, e.g., predefined called numbers and lists of bypass telephone

numbers (BPN) and telephone exchanges, that can be programmed into the gateway during its configuration. As such, local calls and calls to "911" and the like which provide no appreciable cost savings, if any, to a calling party (or his(her) organization) are automatically routed to the PSTN for the entire duration of each such call.

BRIEF DESCRIPTION OF THE DRAWINGS

The teachings of the present invention can be readily understood by considering the following detailed description in conjunction with the accompanying drawings, in which:

FIG. 1 depicts a simplified high-level block diagram of a network environment that incorporates the present inventive VoIP telephony gateway;

FIG. 2 depicts a hardware block diagram of the inventive gateway, e.g., gateway **200**, shown in FIG. **1**;

FIG. 3 depicts a very high-level block diagram of the software that is employed in the inventive gateway;

FIG. 4A depicts a block diagram of an H.323 reference model (operating environment) in which the inventive gateway operates;

FIG. 4B depicts a variant of the reference model shown in FIG. 4A which employs peer border elements according to our present inventive teachings;

FIG. 5 depicts a lower-level block diagram of call processing software **500**, that forms a portion of gateway software **300** shown in FIG. 3 and which is employed in the inventive gateway;

FIG. 6 depicts table **600** showing execution priorities of processes that are used within call processing software **500** shown in FIG. **5**;

FIG. 7 depicts a block diagram of gatekeeper **700** that forms part of call processing software **500** shown in FIG. **5** and also shown as each of gatekeepers 420_1, 420_2, 420_3, 460_1 and 460_2 in FIG. 4B;

FIG. 8 depicts a block diagram of call handler process **560** that forms part of call processing software **500** shown in FIG. **5**;

FIG. 9 depicts a block diagram of border element **900** that forms part of call processing software **500** shown in FIG. **5**;

FIG. 10 depicts a state diagram for peer border element manager **960** shown in FIG. **9**;

FIG. 11 depicts highly-simplified operational sequence **1100** for processing, in accordance with our present invention, a VoIP call between two H.323 telephony endpoints;

FIG. 12 depicts basic inter-process operations **1200** for routing a telephone call over a data network connection (PBX-IP-PBX) between two peered gateways, e.g., gateways **200** and **200'** shown in FIG. 1, located in two different zones;

FIG. 13 depicts typical inter-process control messaging that, in accordance with our invention, occurs both between and within peered gateways, e g., gateways **200** and **200'** shown in FIG. **1**, for routing a telephone call over a data network connecting these gateways;

FIG. 14 depicts typical inter-process control messaging that, in accordance with our invention, occurs both between and within peered gateways, such as gateways **200** and **200'** shown in FIG. 1, for routing a telephone call over a PSTN connection between these gateways;

FIG. 15 depicts typical inter-process control messaging that, in accordance with our invention, occurs both between and within peered gateways, e.g., gateways **200** and **200'** shown in FIG. **1**, for routing a telephone call over the data network but in the absence of a CONNECT message being delivered to a calling side;

FIG. 16 depicts typical inter-process control messaging that, in accordance with our invention, occurs both between and within peered gateways, such as gateways **200** and **200'** shown in FIG. **1**, for switching a telephone call from being routed over a data network connection that spans these gateways to a PSTN connection between these two gateways, and specifically where the latter connection was established through use of a pooled directory number;

FIG. 17 depicts typical inter-process control messaging that, in accordance with our invention, occurs both between and within peered gateways, e.g., gateways **200** and **200'** shown in FIG. **1**, for switching a telephone call from being routed over a data network connection that spans these gateways to a PSTN connection between these two gateways, and specifically where the latter connection was established through use of a called directory number;

FIG. 18 depicts typical inter-process control messaging that, in accordance with our invention, occurs both between and within peered gateways, e.g., **200** and **200'** shown in FIG. **1**, for switching a telephone call from being routed over a PSTN connection that spans these gateways to a data network connection between these two gateways;

FIG. 19 depicts a sequence of inter- and intra-gateway operations **1900** that occurs, in accordance with our invention, for routing a telephone call over data network **30** (as depicted in FIG. 1) between two administrative domains in an H.323 environment, e.g., that shown in FIG. **4B**, where the routing information for the called endpoint has been cached within and is supplied by the border element within the same domain as the calling endpoint ("simple call routing");

FIG. 20 depicts sequence **2000** of inter- and intra-gateway operations that occurs, in accordance with our present invention, for routing a telephone call over data network **30** (as depicted in FIG. 1) between two administrative domains in an H.323 environment, similar to that shown in FIG. **19**, but where the routing information for the called endpoint does not reside in the same domain as the calling endpoint ("complex call routing");

FIG. 21 depicts inter-process messaging that occurs between a gatekeeper and a border element, in the same administrative domain, for establishing a service relationship therebetween;

FIG. 22 depicts inter-process sequence **2200** that occurs to transfer routing information from one gatekeeper to another in the same administrative domain;

FIG. 23 depicts inter-process interactions **2300** that occur between two gatekeepers for tearing down a VoIP call;

FIG. 24 depicts inter-process interactions **2400** that occur, in accordance with our invention, in the course of registering a gateway with a gatekeeper;

FIG. 25 depicts inter-process interactions **2500** that occur, in accordance with our invention, in the course of de-registering a gateway from a gatekeeper;

FIG. 26 depicts inter-process interactions **2600** that occur within a gatekeeper to route a VoIP call being made by a gateway registered with that gatekeeper;

FIG. 27 depicts inter-process interactions **2700** that occur within a gatekeeper to tear down a VoIP call;

FIG. 28 depicts inter-process interactions **2800** that occur within a gatekeeper for registering a new gateway with that gatekeeper; and

FIG. 29 depicts inter-process interactions 2900 that occur within a gatekeeper for de-registering a gateway from that gatekeeper.

To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to various figures.

DETAILED DESCRIPTION

After considering the following description, those skilled in the art will clearly realize that the teachings of the present invention can be readily utilized in a telephony gateway for use in conjunction with any wide area network (WAN), whether it be a private data network or a publicly accessible network, such as the Internet. Our invention is particularly, though not exclusively, suited for use with gateways that are intended to be used with those WANs that rely on the Internet Protocol (IP) to control message routing. Nevertheless, after considering the following discussion, those skilled in the art will readily appreciate how to use our inventive gateway with any of a wide range of differing types of computer networks, other than just IP networks, and to modify that gateway, as necessary, to conform to the requirements of the specific network protocol(s) to be used in any given situation. To simplify the discussion and facilitate understanding, we will describe our inventive gateway in the context of use with a private IP data network. Also, the term "voice" as used herein is generically defined to encompass all types of communication modalities that are typically carried over a subscriber (plain old telephone -- POTs) connection, such as, e.g., speech, facsimile or modem data.

To facilitate reader understanding, we will first provide a brief overview of IP telephony, particular as implemented through the present invention, followed by a description of the hardware components of our inventive gateway, and then followed by the software executed by that gateway. Since our inventive gateway is intended to function in a conventional H.323 environment with H.323 processing being software implemented, then, to provide an appropriate backdrop for the software, we will briefly describe that environment prior to discussing the software in detail. To enhance elucidation, we will then discuss messaging which our inventive gateway implements to provide inter-gateway call routing and associated call handling procedures, including registration and de-registration.

A. Overview

FIG. 1 depicts a simplified high-level block diagram of a network environment that incorporates the present inventive VoIP telephony gateway.

As shown, this environment includes a conventional private IP data (packet) network 30 that inter-connects, via routers 18 and 48, two illustrative Ethernet-based local area networks (LANs) 15 and 45, respectively (while a private network often interconnects a considerable number of separate LANs, for simplicity, only two such LANs are explicitly shown and discussed herein). Each of these LANs itself inter-connects a number of locally situated conventional IP-based devices, such as networked computers, printers and other equipment all of which are not shown for simplicity. LANs 15 and 45 may be widely spaced from each other by a considerable distance, such as one LAN interconnecting such devices situated at a customer site (denoted "Location 1") in one city, such as New York, and another LAN interconnecting additional such devices situated at another site (denoted "Location 2") for the same customer but in a geographically disparate city, such as London, though these LANs need not be so widely spaced apart.

In addition, each location is equipped with various telephones, of which telephone 16 in Location 1 and telephone 46 in Location 2 are illustrative, to serve the individuals stationed there. The telephones that serve any one location are typically connected to a conventional private branch exchange (PBX) which, by performing incoming call termination and outgoing line selection, shares use of telecommunications lines and trunks provided through a local central office (not shown for simplicity) that forms part of public switched telephone network (PSTN) 20, thus reducing cost. PBX 14 is connected to telephones existing in Location 1; PBX 44 is connected to telephones existing in Location 2. For simplicity, only one telephone is shown at each location, though in actuality each location may contain tens, hundreds, thousands or more different subscriber telephones interconnected to a respective PBX. Ordinarily, each of these PBX would be connected, via various outgoing and incoming trunks, to a corresponding telco (telephone company) central office located within and at a fringe of PSTN 20 to route calls over this network. Typically, for a call transiting between Location 1 to Location 2, a user stationed at, e.g., telephone 16, would dial a number of a telephone, such as telephone 46 for an individual, at Location 2. PBX 14 would select an outgoing telephone line to a central office switch (for a relatively large PBX, this amounts to selecting an available time slot in an outgoing, e.g., T1 trunk) and also provide dial tone to telephone 16. The PBX would then pass the dialed number to the central office which then routes the call, via PSTN 20, to a telco central office that serves PBX 44 and, via an incoming trunk that serves that PBX, to a subscriber line terminated by telephone 46. PBX 44 would ring telephone 46 and when an off-hook connection occurs would connect the remote caller through to a called party situated at that telephone.

As thusfar described and as conventionally occurs in many large organizations, telephony traffic does not flow over the data network.

Network 30 and individual LANs 15 and 45 connected thereto are often designed to provide substantial amounts of available bandwidth that accommodates anticipated growth in network usage and which greatly exceeds current user requirements. As such, private network 30 and its interconnected LANs exhibit significant amounts of unused (excess) installed bandwidth which, if not consumed, would be wasted. Hence, network 30 and its LANs can carry some amount of additional traffic over this available bandwidth at what is essentially no additional cost.

To take advantage of this available bandwidth, an inventive gateway is situated on each LAN and interposed between each PBX and the PSTN. In particular, gateways 200 and 200' (which are identical, apart from internally-stored configuration information) reside as distinct ports on LANs 15 and 45, and are situated between PBX 14 and the PSTN, and between PBX 44 and the PSTN, respectively.

Advantageously, each gateway provides either of two paths for telephony traffic to follow: either conventionally through a PSTN, e.g., PSTN 20, or over a data network, such as network 30.

As described in detail below, each gateway dynamically measures quality of real-time connections available through the data network to its peer gateway locations. If the quality of service (QoS) as measured in terms of latency, phase jitter and lost packets, is sufficiently high to support voice traffic, an originating gateway, which serves a calling number, will translate a called number into a corresponding IP address and will route a telephone call through the data network in lieu of the PSTN. Alternatively, if at the time of the call, the

QoS of the data network is inadequate to support high quality speech, the originating gateway will route the call through the PSTN for conventional carriage therethrough to the called party.

In accordance with our inventive teachings, once a telephone call has been initially routed to either PSTN **20** or to data network **30**, then, should the QoS of the data network change, the call will be switched to the other network, with the call having then been dynamically switched, during the duration of the call and in a substantially transparent manner to the calling and called parties, alternately between the data network and the PSTN in response to changes in the QoS of data network **30** such that the call travels over that network then providing a sufficient QoS. Hence, assume that the originating gateway, e.g., gateway **200**, acting in concert with its peer gateway, e.g., gateway **200'**, were to route a telephone call originating at telephone **16** and destined to telephone **46**, over data network **30**. If once the call is so routed and during that call, the QoS of network **30** dynamically decreases, as a result of, e.g., dynamic network congestion which then increases packet latency through this network, to a point which no longer supports high quality voice, then the two gateways will effectuate a telephony connection through the PSTN and then transfer the call from the data network to the PSTN, over this connection. If, while the call is being carried over the PSTN, the QoS of the data network were to return to its proper level, then these peered gateways will establish a data connection between themselves through the data network and switch the call from the PSTN back to the data network over this data connection. Hence, as the QoS of the data network changes during the pendency of a telephone call, the peered gateways will switch the call back and forth between the PSTN and the data network, as needed, to provide high quality end-to-end voice connections at low transport cost.

Not only can the inventive gateway handle and cost-effectively switch voice calls between the PSTN and data network, as necessary, but also the gateway can similar switch other types of calls, such as analog data and facsimile, that are often transmitted on a telephonic connection between two sites.

Furthermore, certain telephone calls, such as calls to emergency numbers (e.g., "911") provide no cost savings to a calling party and hence are best handled through the PSTN, typically the local central office. Similarly, no appreciable cost savings, if any, will arise in by-passing the telephone network for those local (e.g., intra-LATA—local access and transport area) calls that are billed, by a local telephone company, on a flat rate, rather than measured (typically in terms of time and distance), service. Thus, these calls should be handled by the PSTN regardless of the state of the data network.

Advantageously, in accordance with our inventive teachings, each gateway provides selective call routing to route, based on the called number, only those of its outgoing calls to the data network that can provide effective cost savings to the calling parties and/or their organizations. This routing is based on called number information, e.g., predefined called numbers and lists of bypass telephone numbers and exchanges, that can be programmed into the gateway during its configuration.

B. Gateway Hardware

FIG. **2** depicts a hardware block diagram of the inventive gateway, e.g., gateway **200**, shown in FIG. **1**.

As depicted, the gateway is a microcontroller based system having flash memory **205**, random access memory (RAM) **210**, multiple digital signal processors (DSPs) **225**, T1/E1 transceivers/framers **260** and **270**, 4-by-4 port TDM (time-division multiplexed) switch **250** and microcontroller **240**, all interconnected via bus **230**.

The microcontroller is also connected, via leads **247**, to conventional Ethernet network transceiver **255**, which in turn, via leads **258**, is connected to a conventional RJ-45 jack on a physical housing (both the jack and housing are not shown) of the gateway. This jack provides a plug connection to an Ethernet LAN. The microcontroller is illustratively an MPC860T RISC (reduced instruction set computing) microcontroller commercially available from Motorola Corporation of Schaumberg, Illinois. This microcontroller advantageously contains an internal Ethernet interface. Hence, the Ethernet network transceiver provides a physical layer connection from the interface to the Ethernet LAN.

T1/E1 framer/transceiver **260** is connected, via leads **263**, relay **265** and leads **268**, via incoming/outgoing T1/E1 trunk connections, to the PSTN. T1/E1 framer/transceiver **270** is connected, via leads **273**, relay **275** and leads **278**, via incoming/outgoing T1/E1 trunk connections, to the PBX. Both of relays **265** and **275** are connected together via leads **267** to provide, typically in the event of a failure condition, a bypass path around the gateway between the PSTN and the PBX. Though not specifically shown, leads **263**, **268**, **273** and **278** simultaneously but separately carry both incoming and outgoing T1/E1 trunks to corresponding transceiver/framers. Both transceiver/framers are identical to each other and are typically implemented by a BT8370 transceiver/framer currently available from Rockwell International, Inc. Moreover, since each transceiver/framer functions in the same manner regardless of whether it is operating at a T1 or E1 transmission rate (T1 and E1 are North American and European transmission channels operating at 1.544 and 2.048 Mb/sec and capable of carrying twenty-four or thirty, respectively, simultaneous 64 kb/sec time-division multiplexed telephony channels and associated signaling information), then, for the sake of simplicity, we will only discuss transceiver/framer **260** and only at a T1 rate. Inasmuch as leads **263**, **267**, **268**, **273** and **278** carry TDM signals from or to their associated trunks, these leads are also denoted as "TDM lines".

Transceiver/framer **260** accepts an incoming T1 TDM telephony serial signal stream, appearing on, e.g., lead (TDM line) **263** and extracts, for each of the multiplexed time slots in that stream, payload information carried over that slot. This payload information may be digitized voice (including facsimile and digitized data) and/or signaling information; the latter being either channel associated signaling (CAS) or common channel signaling (CCS). For each slot carried by that stream, the associated payload information is then applied, via TDM bus **253**, to a common input port of TDM switch **250**. Similarly, though in reverse fashion, this transceiver/framer also accepts outgoing payload information, from the same port on switch **250**, for each time-multiplexed slot in an outgoing serial stream and assembles that stream, including adding framing and non-payload information, into a T1 TDM signal. Since the transceiver/framer is conventional as is both CAS and CCS, we will omit further details of its operation as well as of these two forms of signaling.

During normal operation of the gateway, the microcontroller applies an appropriate control signal (not shown) to hardware drivers for each of relays **265** and **275** to move their armatures from normally-closed positions to normally-open positions. This, in turn, connects TDM lines **268** to TDM lines **263**, hence applying the incoming and outgoing T1 PSTN trunks to T1/E1 transceiver/framer **260**, and

connects TDM lines **278** to TDM lines **273**, hence applying the incoming and outgoing T1 PBX trunks to T1/E1 transceiver/framer **270**. In the event of a failure condition, as detected by microcontroller **240** or expiration of a watchdog timer (not specifically shown in FIG. **2** but contained within the microcontroller)—the latter indicating a catastrophic failure in the gateway, relays **265** and **275**., which are electromechanical in nature, will both assume their normally-open position. In this position, TDM lines **268** will be connected through relays **265** and **275** directly to TDM lines **278**, hence connecting the PSTN T1 trunks directly to the PBX T1 trunks and effectively bypassing the gateway.

TDM switch **250** is illustratively formed by model MT8981D 4-by-4 digital switch commercially available from Mitel Corporation. Each of the inputs to the switch can accept up to 32 separate 64-kb/sec channels multiplexed onto a 2.048 Mb/sec (E1) stream. Each of the outputs provides a serial output TDM signal at the same rate as the input. The switch, operating under the control of microcontroller **240**, provides "switched" time slot connections and can write data into a desired time slot in a TDM signal appearing at one of the serial outputs from the switch, where the data can originate from either a desired time slot in any of the four serial TDM input signals applied to the switch or from the microcontroller itself. Similarly, the microcontroller can read, through the switch, the signal value appearing in any desired time slot of any of the four serial inputs.

In essence, this switch provides appropriate time-slot connections between TDM signals appearing at its inputs and outputs in order to either route a telephone call on a given TDM channel between the PSTN and the PBX, thus by-passing the private data network, or route that call to the DSP, for appropriate processing, and ultimately the microcontroller for carriage over the private data network.

Specifically, a signal on a channel in an incoming T1 trunk, such as that carried by TDM lines **268**, and originating from the PSTN, can be switched, through switch **250**, to a corresponding time slot on an outgoing T1 trunk, such as over TDM lines **278**, to the PBX, and vice versa, in order to support carriage of that call over the PSTN between caller and called locations. Such a telephony connection (denoted as "PBX-PSTN") is represented by dashed line **290**. Such a channel will be processed through both T1/E1 transceiver framers, first to extract a signal on that channel from an incoming T1 TDM trunk, by one of the transceiver/framers and then, through the other transceiver/framer, to assemble the proper outgoing T1 TDM trunk.

Alternatively, if the gateway were to route an outgoing telephony call from a calling device, such as a telephone, computer modem or facsimile machine, connected to the PBX over the private data network (to effectuate a "Voice over IP" or VoIP call) instead of the PSTN, TDM switch **250**, based on control information provided by microcontroller **240**, connects an incoming time slot for that call, not to a time slot via T1/E1 transceiver/framer **2** and, from there, to an outgoing T1 trunk, but rather, via TDM bus **228**, to an input of a DSP then available within DSPs **225** and ultimately to microcontroller **240**. Collectively, that DSP and the microcontroller convert the digitized telephony signal for that call into suitable IP packets and transmit those packets, with appropriate IP addresses, over the LAN for subsequent carriage over the data network to a peer gateway. The peer gateway receives these IP packets, via the data network and its LAN, and, for each such call, performs the reverse operations to convert these packets back into a digitized telephony signal for that call and then routes this signal to a destination PBX for conversion back to an analog telephony signal and ultimately termination of that signal to a called telephone or other analog telephonic device, such as a computer modem or facsimile machine. Each separate called number has an associated IP address, which ultimately is known to both peered gateways—as will be described in detail later, such that the peered gateways can properly address the IP packets to their unique called destination.

DSPs **225** illustratively contain eight separate DSPs $225_1, \ldots, 225_8$ (six of which can collectively implement 24-channel T1 operation, with eight collectively implementing 30-channel E1 operation). Each DSP, which is illustratively a model TMS320C549 DSP commercially available from Texas Instruments of Dallas, Tex., handles four simultaneous channels of digitized telephony traffic as provided by the PBX. SPAM (static random access memory) **220** contains SRAMs $220_1, \ldots, 220_8$, with each separate SRAM providing temporary data storage for a corresponding and different DSP.

In particular, for a digitized signal appearing on any one such TDM channel (a single time slot), such as that incoming from the PBX to the gateway, a DSP assigned to that channel first converts a G.711 compressed telephony signal (typically between 56–64 kb/sec) for that channel and provided by the PBX into a G.723 compressed signal (typically between 5.6–6.4 kb/sec) to effectuate a 10:1 compression. Currently, the gateway relies on use of a "digital PBX" i.e., the PBX provides, for each of its outgoing channels, analog-to-digital conversion (digitization) and compression, according to a G.711 standard, and, for each of its incoming channels, G.711 decompression and digital-to-analog conversion. If a digital PBX were not used, then appropriate channel banks could be added between the PBX and the gateway to provide these functions.

Once G.723 compression is effectuated for any channel, the DSP suitably packetizes the G.723 compressed signal. Resulting G.723 packets are then applied, via bus **230**, to the microcontroller. The microcontroller, upon receipt of each of these packets for any one channel, assembles these packets for that channel into proper IP packets with necessary IP headers, including originating and destination IP addresses as well as other required information, and transmits these packets, via its internal Ethernet interface and Ethernet network transceiver **255**, to the LAN for subsequent routing to a peer gateway. Such a telephony connection (denoted as "PBX-IP") carried over the private data network is represented by dashed line **295**. Inasmuch as G.711 and G.723 compression algorithms are well known in the art, we will omit all details of these algorithms.

To process a VoIP call appearing on the LAN and incoming from the private data network to a called directory number serviced by the PBX, microcontroller **240**, upon receipt of IP packets via network transceiver **255**, first extracts a destination IP address and payload packetized telephony data from each of these incoming IP packets. The microcontroller, as discussed in detail below, determines from its internal routing tables, a correspondence between that destination IP address and that called number. Once this number is ascertained, the microcontroller establishes, through switch **250**, a time-slot connection, via an outgoing TDM trunk, e.g., that appearing on TDM lines **273** and **278**, to the PBX, to a TDM channel that will serve that called number. The microcontroller applies each telephony packet appearing in each of the IP packets to an available DSP which, in turn, de-packetizes that packet into G.723 compressed data and converts the G.723 data into G.711 compressed data. The resulting G.711 data is inserted, through

the time-slot connection established through the switch, into a specific channel in an outgoing TDM signal destined to the PBX and specifically that which will be connected by the PBX to the called number. The TDM signal is then applied to transceiver/framer **270** which, in turn, assembles a proper T1 signal and thereafter, applies that signal, via TDM lines **273** and **278**, to the PBX.

In those instances where the T1 trunks utilize CCS (ISDN Primary) mode signaling, microcontroller **240** instructs switch **250** to either extract D (data) channel signaling information from incoming TDM signals or insert D channel signaling information into outgoing TDM signals. In that regard, for an incoming TDM signal appearing on either TDM bus **253** or **256**, switch **250** extracts this signaling information and, via TDM bus **243**, routes that information to the microcontroller for use in subsequent call processing. In a reverse fashion, for an outgoing TDM signal generated by the switch and appearing on either TDM buss **253** or **256**, the microcontroller generates appropriate D channel signaling information and applies that information via TDM bus **243** for proper insertion by the switch into that outgoing TDM signal for proper downstream call processing.

Flash memory **205**, illustratively 4 Mbytes in size, stores program code and other information, such as call routing (translation) tables, in a non-volatile manner. The gateway includes appropriate circuitry (not shown), along with software processes, through which the contents of the flash memory can be updated, as needed. SDRAM (synchronous dynamic random access memory) **210** is illustratively 2 Mbytes in size. Not only does the SDRAM provide temporary data storage, it also receives, during system start, a copy of the program code stored within the flash memory. This program, as noted below, is then executed out of the copy stored in the SDRAM.

C. Software and H.323 environment

1. Software overview—operating system and call processing software

FIG. **3** depicts a very high-level block diagram of gateway software **300** that executes in our inventive gateway. As shown, this software, at its highest level, constitutes conventional operating system (O/S) kernel **310** and call processing software **500**.

The O/S kernel provides, among other services, process creation, scheduling, inter-process communication and event signaling. Inasmuch as details of the operating system are not relevant to the present invention, all such details will be omitted from the ensuing discussion.

Call processing software **500** is formed of a number of distinct processes and drivers, as shown in detail in FIG. **5** and discussed below, which collectively implement VoIP call processing in an H.323 environment.

2. H.323 environment

FIG. **4A** depicts a block diagram of an H.323 reference model (operating environment) in which the inventive gateway operates.

In general, an H.323 network is one which utilizes a packet-based network, as its transport vehicle, but which may not provide a guaranteed QoS. Such packet based networks may include LANs, enterprise area networks, metropolitan area networks, intra-networks (intranets), such as a private data network, and inter-networks (including the Internet). These networks also include dial-up connections or point-to-point connections over, e.g., the PSTN, or an ISDN connection which uses an underlying packet based transport such as PPP (point-to-point protocol). These networks may consist of a single network segment, or they may have complex topologies which incorporate many network segments interconnected by other communications links. To the extent relevant to the present invention, H.323 terminals (here specifically denoted, for the purpose of telephony communications, as H.323 "telephony endpoints") provide audio, facsimile and/or data communications capability for point-to-point communications. Interworking, over an H.323 network, between H.323 endpoints is accomplished through so-called "gateways", such as gateways **200** and **200'** (also see FIG. **2**), which provide, inter alia, admission control and address translation services.

An H.323 endpoint can be any telephony or network-connectable device that is compliant with the H.323 standard. An endpoint can initiate a call to another such endpoint as well as be called by another endpoint. In general, an endpoint generates and/or terminates information streams. An endpoint can be a telephone (being a "telephony endpoint") or other customer premise device (CPE), provided it contains appropriate circuitry or is interfaced to appropriate circuitry, such as a H.323 PBX, that provides H.323 compliance.

Each telephony endpoint has an alias address, in the form of a directory number—as illustratively specified in the H.225 standard, associated with it. For any one telephony endpoint, its alias represents an alternate method of addressing it and is used as a method of internetworking with that telephony endpoint through the PSTN. Gatekeepers, on the other hand and discussed below, do not possess any such alias.

An H.323 network, primarily for telephony use, as illustrated in FIG. **4A**, consists of a number of zones (four of which are illustratively shown, specifically zones **405**, **410**, **470** and **475**) and administrative domains (two of which are illustratively shown, specifically Administrative Domains A and B). For our purposes, we view a zone as a group of H.323 telephony endpoints which are controlled, for purposes of their effectuating interconnections, through a single gatekeeper. Here, for example, zone **405** contains H.323 endpoints $\mathbf{405}_1, \ldots, \mathbf{405}_w$ all controlled by gatekeeper $\mathbf{420}_1$; zone **410** contains H.323 endpoints $\mathbf{410}_1, \ldots, \mathbf{410}_x$ all controlled by gatekeeper $\mathbf{420}_2$; zone **470** contains H.323 endpoints $\mathbf{470}_1, \ldots, \mathbf{470}_y$ all controlled by gatekeeper $\mathbf{460}_1$; and zone **475** contains H.323 endpoints $\mathbf{475}_1, \ldots, \mathbf{475}_z$ all controlled through gatekeeper $\mathbf{460}_2$ (where w, x, y and z are integers). In essence, a gatekeeper is a logical H.323 entity that provides IP call routing functions; a gateway converts between circuit-switching calls and VoIP calls.

Generally speaking, an administrative domain contains a set of H.323 entities that are administered by one administrative entity, within the domain. Each H.323 entity in the domain has at least one network address that uniquely identifies that entity. Here and in the context of telephony applications, each administrative domain contains a group of zones that are isolated from other zones, with each zone being managed by a separate corresponding gatekeeper. As shown, Administrative Domains A and B contains zones **405** and **410**, and **470** and **475**, respectively. An administrative domain provides call routing services for its zones through gatekeeper-to-gatekeeper messages or gatekeeper-to-border element messages.

A border element is a functional element which provides a path to accommodate external, i.e., public, access into an administrative domain for purposes of call completion or any other service which involves multimedia communication with any other element located within the administrative domain. Such access into a domain occurs strictly through a border element, Hence, a border element effectively controls an external view of its domain. A border element commu-

nicates with other border elements, such as for inter-domain communication for, e.g., inter-domain call completion, through use of a protocol specified in "Annex G" to the H.225 standard. Border element-to-border element messaging provides the sole vehicle through which a border element in one domain can establish a service relationship with a border element in another domain in order to complete a call between these domains. Within any one administrative domain, a border element within that domain may communicate, using H.225 signaling, with other H.323 elements, such as gatekeepers located within that domain. A border element within any one domain is responsible for diffusing domain-wide routing information supplied to it by its associated gatekeeper, i.e., routing information that involves any endpoint (telephony endpoints and gateways) in that domain, such as registration (or de-registration) of a new gateway or telephony endpoint and/or a change in a translation table, to all other gatekeepers throughout that domain, thereby ensuring that routing tables stored at each gatekeeper in the domain and in the border element are consistent across all gatekeepers and providing redundancy across gatekeepers. Moreover, since each gatekeeper locally maintains a complete, constantly updated copy of this information, network processing speed is increased by eliminating any need to centralize this information and hence any need, with its attendant processing and network delays, to query a centralized database for any of this information.

As shown, Administrative Domains A and B contain corresponding border elements **430** and **450**. As shown, border element **430** can communicate with gatekeepers **420**, including gatekeepers $\mathbf{420}_1$ and $\mathbf{420}_2$; while border elements **450** can also communicate with gatekeepers **460**, including gatekeepers $\mathbf{460}_1$ and $\mathbf{460}_2$. Communication between H.323 elements, such as gatekeepers, located within different administrative domains only occurs through their associated border elements. Hence, communication between gatekeepers **4202** and $\mathbf{460}_1$ would be handled by and pass through both border elements **430** and **450** and occur in accordance with the Annex G standard. Messages between border elements, such as elements **430** and **450**, are typically carried over UDP. So-called "keep alive" messages are exchanged between the border elements, including "peer" border elements as described below, such that each border element will continually detect whether another border element has failed. H.225.0 messages that are exchanged between border elements, e.g., elements **430** and **450**, include: information download and information update. These messages will be discussed below in their context of use.

Gatekeepers establish service relationships with border elements, in their common domain, to acquire and publish call routing information therebetween within an administrative domain. Further, a border element, e.g., border element **430**, in response to a request from a gatekeeper, e.g., gatekeeper $\mathbf{420}_1$, in the same domain as the border element can request routing information for calls outside its zone and its domain from a border element associated with a domain containing a called telephony endpoint.

Gatekeeper-to-gatekeeper messages are represented by line **413** for messages between gatekeepers $\mathbf{420}_1$ and $\mathbf{420}_2$ and line **465** for messages between gatekeepers $\mathbf{460}_1$ and $\mathbf{460}_2$. Gatekeeper-to-border element messages are represented by lines **423** and **427** for messages between border element **430** and gatekeepers $\mathbf{420}_1$ and $\mathbf{420}_2$, respectively; and by lines **453** and **457** for messages between border element **450** and gatekeepers $\mathbf{460}_1$ and $\mathbf{460}_2$, respectively. Messages between border elements and gatekeepers are passed typically using UDP (user datagram protocol). Here too, "keep alive" messages are exchanged between each gatekeeper and its corresponding border element such that both elements will continually detect whether the other has failed. H.225.0 messages that are exchanged between gatekeepers and a border element include: service request/service confirm/service reject; descriptor ID request/descriptor ID confirm/descriptor ID reject; descriptor request/descriptor confirm/descriptor reject; and descriptor update/descriptor update acknowledge. These messages will also be discussed below in their context of use. Border element functionality, as will be discussed in conjunction with FIG. **5** below and with other successive figures, being software implemented may exist in combination within other H.323 elements, such as a gateway or even a gatekeeper.

Each gatekeeper itself is an H.323 entity on the network that principally provides address translation and controls access to the network for other H.323 devices, such as H.323 telephony endpoints and other gateways. A gatekeeper is logically separate from the telephony endpoints; however, its physical implementation may coexist with a terminal, gateway or other non-H.323 network device.

In particular, each gatekeeper provides address translation by translating between an Alias Address (phone number) of a telephony endpoint and its network transport address (IP address). This translation is effected through a translation table, which is updated using registration messages and other methods, as described below. Further, each gatekeeper provides admissions control by authorizing network access, through H.225.0 messages, based on, e.g., call authorization, network bandwidth required and available, or other criteria, and also controls bandwidth by regulating bandwidth allocated to any call transported over the data network. Each gatekeeper provides these functions for both telephony endpoints and gateways that have registered with that gatekeeper, thus effecting zone management.

H.225.0 messages that involve a gatekeeper include: gatekeeper request/gatekeeper confirm/gatekeeper reject; registration request/ registration confirm/registration reject; admission request/admission confirm/admission reject; disengage request/disengage confirm/disengage reject; information request/information request response/information request acknowledge/information request nack (no acknowledge); and bandwidth request/bandwidth confirm/bandwidth reject.

Furthermore, each gatekeeper can provide various optional functions, as needed. One such function is call control signalling. Here, the gatekeeper may choose to process call signalling information that occurs between two telephony endpoints themselves and thus complete the signalling rather than handing that function off to the another element. Alternatively, the gatekeeper may direct the two telephony endpoints to connect a call signalling channel directly to each other, thus avoiding a need for that gatekeeper to handle H.225 call control signals. Another such optional function is call authorization. Specifically, through the use of the H.225 signalling, a gatekeeper may reject calls from a telephony endpoint due to an authorization failure. The reasons for such a rejection may include, e.g., restricted access to/from particular telephony endpoints or gateways, and restricted access during certain periods of time. An additional optional function is bandwidth management. Here, a gatekeeper controls a number of H.323 telephony endpoints permitted simultaneous access to the data network. Through the use of the H.225 signalling, the gate-

keeper may reject calls from a telephony endpoint due to bandwidth limitations. This may occur if the gatekeeper determines that insufficient bandwidth is available on the data network to support a requested call. This function operates during an active call whenever a telephony endpoint requests additional network bandwidth. Furthermore, a gatekeeper provides desired call management functionality. This includes, e.g., maintaining a list of ongoing H.323 calls. Information contained in this list may be necessary to indicate that a called telephony endpoint is busy and/or to provide information for the bandwidth management function. Lastly, a gatekeeper can reserve network bandwidth for telephony endpoints that lack this function and can provide directory services to access and supply directory information for translation between aliases and network transport addresses.

Back-end services **440** are available, via network links, to various H.323 entities located within the domains and provide various centralized functions needed to support call completion throughout the H.323 environment. These functions include user authentication or authorization, accounting, billing, rating/tariffing and other administrative and support functions best provided on a centralized basis from a remote server or data center. Furthermore, back-end services **440** can also provide call routing information to requesting entities for those destination endpoints that can not be resolved on an inter-domain basis.

As shown, back-end services **440** can communicate with Administrative Domain A through links **442** and **444** to gatekeeper $\mathbf{420}_1$ and border element **430**, respectively, and with Administrative Domain B through links **446** and **448** to gatekeeper $\mathbf{460}_1$ and border element **450**, respectively. Back-end messages to gatekeeper $\mathbf{420}_2$ and $\mathbf{460}_2$ will be accommodated through inter-gateway messaging carried over link **413** and **465** from gateway $\mathbf{420}_1$ to gateway $\mathbf{420}_2$, and from gateway $\mathbf{460}_1$ to $\mathbf{460}_2$, respectively; as well as in a reverse direction.

As defined in the H.323 standard, an H.323 telephony call begins with a specific call set-up procedure and ends with a specific call termination procedure—both of which are discussed in detail below.

Inasmuch as the telephony endpoints, such as, e.g., endpoints **405**, **410**, **470** and **475**, are to inter-network with a switched telecommunications network, e.g., the PSTN, rather than just confine their operation to an IP data network, each call involving one of these endpoints terminates at an appropriate gateway where data for that call is converted, as discussed above in the context of FIG. **2**, to a proper representation for carriage over a desired network, either the PSTN or the private data network.

Our inventive gateway uses standardized call signaling protocols and packetization as defined in the H.225 standard. Inasmuch as the H.225 and H.323 standards are well-known in the art, for the sake of brevity, we will omit a detailed discussion of the specific procedures and messaging undertaken in the gateway and the endpoints that are implemented as specified in and hence compliant with these standards.

In accordance with our present inventive teachings, we have extended the H.323 reference model to incorporate peered border elements within a single administrative domain. These border elements provide increased fault tolerance and redundancy. FIG. **4B** shows such an extended reference model. Since most elements in this extended reference model are identical with those shown in FIG. **4A**, we will only focus on those that are additional. Inasmuch as the network topologies shown in FIGS. **4A** and **4B** are merely illustrative, those skilled in the art will realize that a wide variety of different topologies of differing complexities which still incorporate the concepts shown in these two figures can be utilized in any actual network implementation.

As shown in FIG. **4B**, administrative domain A includes border element **430** (also included in the reference model shown in FIG. **4A**) and its peer border element **430'**. Though preferably two border elements can be established as "peers", more than two can be utilized as well. Since presently, two border elements are used as "peers" to simplify configuration and testing, we will limit the ensuing discussion to that particular configuration. However, it will be readily apparent to anyone skilled in the art how to extend the architecture of a domain to include an increased number of border elements all operating as "peers" of each other. Though any or all of the administrative domains can include peered border elements, we will also limit our discussion to only one such administrative domain.

Peered border elements **430** and **430'** function together and behave as a single monolithic border element, i.e., one "logical" border element, but with their functionality being duplicated across both such elements. Hence, if either border element fails, the other can provide inter-domain routing and inter-zone routing within a common administrative domain. Peer border element **430'** also handles communication, for gatekeeper **4303** for additional zone **415**, i.e., zone **415** which contains telephony endpoints $\mathbf{415}_1, \ldots, \mathbf{415}_m$ (where m is an integer). Peered border elements preferably have a loosely coupled distributed architecture, with no hierarchical differences. Such border elements, in a "peered" relationship, are not operated in a master/slave or active/standby basis. Transactions from gatekeepers or one border element in a domain, e.g., such as border element **430** functioning as an external border element (i.e., it provides external access into the domain), are shared with its peer border element, here **430'**. As such, transaction data stored in one peered border element remains synchronized with that stored in the other such that either one border element can immediately undertake transaction processing should its peer border element fail or be taken out of service. As indicated in FIG. **4B**, peered border elements **430** and **430'** in a common administrative domain establish TCP/IP channels between themselves, with two connections existing: one originating at each such element and terminating at the other. Hence, each border element in a "peered" configuration has both TCP/IP server and client connections. H.225.0 messages between peer border elements, similarly to those between external border elements (such as elements **430** and **450**), include information download and information update message, as well as to establish and disconnect TCP/IP connections therebetween.

The information download message is sent by one "originating" border element to its peer, upon establishing a TCP connection with that peer. This message shares all the call routing capabilities of the originating border element with its peer. The message contains local service relationships (internal to a domain), local descriptors, external service relationships (external to a domain) and external descriptors. The local service relationships define transport addresses of each of the gatekeepers that have a service relationship with the originating border element. The local descriptors define, in accordance with the Annex G standard, routing descriptors and are obtained from either a static configuration of the same domain as the originating border element or from gatekeepers, located within this domain, that have established service relationships with the originating border element. The external service relationships define, for the

originating border element, here, e.g., element **430**, transport addresses of those border elements external to this domain, such as border element **450** which resides outside of administrative domain A, that have established a service relationship with the originating border element. The external descriptors define, in accordance with the H.225 standard, routing descriptors, that are obtained from either a static configuration of the H.323 environment or from border elements, located external to the domain that contains the originating border element, that have established service relationships with the originating border element. An information update message is sent from the originating border element to its peer in order to notify the latter of a change either in information affecting a gatekeeper located within the same domain or information received from a border element located external to this domain. The particular border element within an pair of "peered" border elements that originally received such information is responsible to send that information to all its peers.

For further details on the H.323 and H.225 standards including Annex G, the reader is referred to for: (a) the H.225 standard: "Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services—Transmission multiplexing and synchronization, Call Signaling Protocols and Media Stream Packetization for Packet-based Multimedia Communications Systems", ITU-T Recommendation H.225.0, draft version 3, May 1999; (b) the Annex G standard: "Annex G—Communication between Administrative Domains" (referred to herein as "H.225"), ITU, Draft of H.225.0 Annex G for decision, 17–28 May 1999; and (c) for the H.323 standard: "Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services—Systems and terminal equipment for audiovisual services, Packet-based Multimedia Communications Systems", ITU-T Recommendation H.323, version 3, May 1999, all of which are incorporated by reference herein. The reader is also referred, for details of the control protocol used in the invention between H.323 elements, such as, e.g., gatekeepers and endpoints, to "Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services—Communication procedures, Control protocol for multimedia communication", ITU-T Recommendation H.245, July 1997—which is also incorporated by reference herein.

3. call processing software **500**—constituent processes and other software

With the above in mind, we will now turn to FIG. **5** which shows a detailed block diagram of call processing software **500** that forms a portion of software **300** used in our inventive gateway. As indicated in a key shown in FIG. **5**, circular and rectangular blocks correspondingly designate process, and drivers and other software modules; thick solid lines denote data paths; thick and thin dashed lines denote signaling and configuration information paths, respectively; and thin solid lines denote paths for other software interactions. We define a "process" as an independent execution entity of which the system operating system is aware. A process can contend for system resources then controlled by the operating system, such as, e.g., the processor, memory and input/output (I/O) access. A process can be divided into multiple tasks, each of which is a logical entity of which the operating system has no knowledge.

Though the inventive gateway uses an event-driven multitasking pipelined software architecture, for simplification, we have intentionally omitted nearly all details of these aspects of the architecture from the ensuing discussion. Such details are conventional in nature; hence, their use in implementing the software used in our present inventive gateway would be readily apparent to any one skilled in the art.

As shown, overall functionality of software **500** can be divided into four basic sections: data section **510** which includes, solely for purposes of illustration separate sections $\mathbf{510}_A$ and $\mathbf{510}_B$; call processing section **550**; call switching section **580** and call signaling section **590**. Data section **510** controls transmission and reception of packetized traffic, including packetized telephony, over the LAN connection to the gateway, as well as converting, for any VoIP call, between its TDM telephony representation, in G.711 compressed form, and its IP data packet representation, with G.723 compression.

Call processing section **550** manages the H.323 environment in which the gateway functions; allocates call handling resources to process a call; and routes telephony calls over either the PSTN or data network and, through interaction with voice packet handler process (VPH) **517** and TASQ process **537** within data section $\mathbf{510}_A$, switches those calls back and forth between the PSTN and the data network, as warranted by QoS then being provided over the data network. Call signaling section **590** generates appropriate telephony signaling information for use by the PSTN or the PBX for properly routing calls, either through the PSTN or over the data network, between the gateway and the PSTN, and between the gateway and the PBX, respectively.

Though not specifically part of any of the sections, software **500** also contains command and manufacturing test library **575**, buffer manager **593**, timer manager **594**, interprocess communication facility **595**, and event log server **596**.

Command and test library **575** provides a library of commands through which a user can interact, such as through a console or personal computer, with the gateway, through either RS-232 driver **539** and a serial port provided on the gateway. Through such interaction, a user can set system configuration parameters, invoke various internal test procedures, and perform other functions provided by the gateway, such as, e.g., reading internal event logs, downloading internal operational statistics, and updating various software modules, such as DSP drivers.

Separate pools (not specifically shown) of data and control buffers are provided in the gateway for dynamic allocation and use by any requesting process. These pools are managed by buffer manager **593**. The control buffer pool, which contains a number of 268-byte buffers, are used to send signaling messages between HDLC (high-level data link controller (D channel)) driver **592** and various processes, specifically, Q.921 process **572**, Q.931 process **577**, T1AB process **575** and call handler process **560**—all of which will be discussed below. In addition, other processes can utilize these buffers for sending inter-process control communication amongst themselves. The data buffers, having a number of 256-byte buffers, are used to transport data messages between Ethernet driver **533** and VPH process **517**. Each of these buffers is sufficiently large to accommodate an RTP header (12 bytes long) and 240 ensuing bytes of voice samples, which, in the absence of using G.711 compression, stores approximately 30 msec of voice samples.

Timer manager **594** provides and manages various software-based timers for various requesting processes and drivers. This manager starts and stops timers, as needed. For processes, manager **594** delivers a Timer Expiry message to a process that had previously set a now expired timer. Device drivers interact with manager **594** through a call back approach inasmuch as device drivers are incapable of

receiving a message. In use, manager **594** receives a timer interrupt every 10 msec during which this manager determines if any active timer has then expired and, if so, delivers identification of that timer in an interrupt service domain.

Inter-process communication facility **595** implements an interface mailbox for each process through which that process communicates with device drivers and other processes. This mailbox consists of a doubly-linked queue head and other information to facilitate inter-process event signaling. A calling process, i.e., a process that is sending a message to, e.g., another or "called" process, sends a message to the latter by identifying a particular message buffer, and placing that identification into a mailbox of the called process. If necessary, an event is generated to "wake up" the called process upon delivery of a message. A message consists of a system message header and a variable length data field. The header specifies message type, a unique system identification (USID) associated with the calling process. Each process and driver has an associated two-byte USID. A USID is associated with each different function, not with a given hardware device, such as a specific circuit. A USID has two single-byte components: a major device number and a minor device number. A device is assigned a major device number, with each different function provided by that device being assigned a different minor device number. Processes are viewed as "virtual" devices, with their one major device number, and each different task associated with that process having a different minor device number. Through this approach, a process can receive a message from another process as well as receive incoming data or an event from a device driver.

Event log server **596** maintains an event log table and provides an event log service. Through this service, certain commands can be used to generate events which will write an entry into, and, hence, update the event log. This log can be read by a user by issuing appropriate Telnet commands which are routed by command and manufacturing test library **575** to Telnet process **526**.

Now, turning to the specific sections of software **500**, we will first address data section 510_A.

As depicted, data section 510_A contains, from the standpoint of processes: idle process **502**, configuration manager (CM) **505**, web server **514**, VPH **517**, HTTP (hypertext transfer protocol) server **520**, FTP (file transfer protocol) server **529**, Telnet server **526**, SNMP (simplified network management protocol) process **538**, and TCP/IP process **535**. This section also includes, with respect to drivers: DSP driver **519**, Ethernet driver **533** and RS-232 driver **539**; and with respect to other software modules: watchdog timer module **507**, database **508**, web pages **511** and flash memory programming module **523**.

In particular, upon power-up of the gateway, a boot program stored in flash memory **205** (see FIG. 2) copies executable program code from the flash memory into SDRAM **210** and then transfers execution to the program copy then residing in SDRAM. Copying the code and subsequently executing it from SDRAM is substantially faster than directly executing that code directly from the flash memory. Once this code begins executing from SDRAM, it initializes various configuration tables, boots the operating system and then passes control to the operating system. The operating system starts configuration manager **505** as a first process to be executed, with it, in turn, spawning all other processes, as needed. The configuration manager also initializes buffer manager **593**, timer manager **594** and event log server **596**. Once this occurs, the configuration manager then initializes all the device drivers and other software modules requiring initialization. Also, upon occurrence of a power-on reset condition, configuration manager **505** clears the event log and all statistics counters. In addition, the configuration manager also starts watchdog timer driver **507**. This driver regularly and periodically strobes (resets) a hardware-implemented watchdog timer situated within the microcontroller to continually re-start its timing interval. In the event of a catastrophic failure condition which halts software execution, the watchdog timer will reach the end of its timing cycle and generate an alarm condition, as well as cause relays **265** and **275** (see FIG. **2**) to assume their normally-open position and hence bypass the gateway.

The configuration manager maintains configuration information within database **508** shown in FIG. **5**. This information contains two basic portions: unit-specific information, i.e., for a specific gateway in which software **500** is executing, and H.323 domain-wide information. Unit-specific information, i.e., a so-called "profile", is separately configured for each gateway operating in the H.323 environment. Domain-wide information can be entered into any one gateway which, in turn, will distribute, using a multicast mechanism, that information to all other gateways in the same administrative domain. Domain-wide information includes, e.g., H.323 registration information of each gateway, border element and telephony endpoint in that domain. As discussed below, as each H.323 element registers with a given gatekeeper, that gatekeeper, through an associated border element, broadcasts that registration data to every other gatekeeper in the same administrative domain such that each gatekeeper and border element in the domain knows the existence and address of every other H.323 element in the domain. The converse occurs as each H.323 element de-registers and leaves its domain. This database can be dynamically updated, to reflect current configuration status, by any of a number of different processes: web server **514**, SNMP process **538**, gatekeeper **700**, call handler **560**, Q.931 process **577** and border element **900**. Database **508** also stores the routing (translation) tables. Both the profile and system-wide information, as well as the routing tables, are stored within flash memory **205** (see FIG. **2**) to provide non-volatile storage across system resets and power cycles.

TCP/IP process **535**, shown in FIG. **5** and which lies at a heart of data section 510_A, implements a basic routing engine within the gateway. Specifically, this process implements a TCP/IP (transmission control protocol/internet protocol) stack with destination based IP routing. This process performs all processing of entries in this stack for IP, TCP, UDP and ARP (address resolution protocol) protocols. A conventional "sockets" interface is provided by process **535** to permit communication, at a top layer of the stack, with local IP applications, specifically: VPH **517**, HTTP server **520**, FTP server **529**, Telnet server **526**, SNMP (simplified network management protocol) **538**, TASQ process **537**, Gatekeeper (GK) **700**, event server **555** and P.323 process **553**. A common network interface, situated at a bottom layer of the stack and communicating with Ethernet driver **533**, facilitates network communication with the stack through an Ethernet (LAN) connection. In particular, process **535** accepts incoming IP packets from the LAN, as supplied by Ethernet driver **533**. In that regard, each of these packets, as conventionally occurs, was encapsulated, as payload data, within an Ethernet packet and is extracted therefrom by the Ethernet driver. As such, process **535** routes the IP packet to either one of the local applications or protocols for processing, based on a protocol ID and well-known port number contained within the packet. Similarly,

process **535** can route IP packets generated by any one local IP application to another for processing by the latter. Since the HTTP, FTP, Telnet and SNMP processes are all conventional, we will not discuss them in any further detail.

As noted above, software updates, such as to a driver or process, can be provided, via user entered telnet commands, through FTP process **529** to the system. Any such update, in the form of replacement code, is written, through use of flash programming module **523**, into flash memory **205** (see FIG. **2**) and, as such, overwrites a corresponding prior version of the code.

Conventional web server **514** shown in FIG. **5**, operating through HTTP server **520**, provides access to pre-defined web pages **511**, with user entry dialogs, through the user can enter configuration information into the gateway. Once entered, web server **514** writes this information into appropriate locations within database **508**. Whenever a user, such as through a browser executing on a personal computer or workstation residing on the LAN, attempts to establish a TCP/IP connection to the gateway, HTTP server **520** starts which, in turn, instructs web server **514** to download an appropriate entry "home" page to that user. Telnet server **526** implements a server side of the Telnet protocol which permits a telnet client, executing on that PC or workstation, to communicate with the gateway. A user interface is provided, via command and manufacturing test library **575**, through which, as discussed above, the user can interact with the gateway, and, e.g., update software through FTP process **529**, download log entries, execute various manufacturing tests (such as T1/E1 framing, loopback, LED tests and others), and so forth. SNMP process **538** is a client SNMP process that encodes and transports network management messages, such as locally generated alarms, to a remote standardized SNMP management platform and permits configuration of the gateway through that platform.

Furthermore, TCP/IP process **535** can also route outgoing IP packets, via Ethernet driver **533**, for carriage over the LAN for external processing. As discussed below, these outgoing IP packets can be locally generated VoIP packets containing near-end voice, data or facsimile information originating with a caller which, in turn, are routed by the LAN to the private data network and from there to a remote gateway for eventual conversion into TDM signals and termination at a destination PBX that serves a called directory number.

For each VoIP call transiting through the gateway, VPH process **517** bi-directionally handles voice packets associated with that call. Specifically, incoming G.711 packets, containing voice data originating in a TDM channel from a local PBX, are provided by call handler process **560**, as shown in FIG. **5**, to VPH process **517**. This latter process provides these packets to DSP driver **519**. This driver, in turn, routes these packets to an allocated channel on an assigned DSP (one of DSPs $225_1, \ldots, 225_8$) for conversion into G.723 compressed form. The resulting G.723 packets are then returned to the VPH which, in turn, encapsulates these packets with appropriate IP call routing information into IP packets. The call routing information includes destination and originating IP addresses, associated with the called and calling directory numbers and defined in the translation table stored within database **508**, accessed by the call handler and supplied to VPH process **517**. Thereafter, VPH process **517** supplies these IP packets to TCP/IP process **535** for routing to the LAN and, from there, to the private data network for carriage to a far-end peer gateway. The VPH processes data packets incoming from the LAN in an opposite fashion, to that described above, to provide corresponding G.711 packets, to the call handler process, for eventual conversion into a TDM channel and application to a local PBX.

DSP driver **519**, through various counters and buffers, also determines packet loss statistics through conventional RTP (real-time transport protocol—which resides within the UDP) packet sequencing numbers contained in the packets and provides buffer over/under-flow information for use in determining jitter of the for each current VoIP call it is then handling. For each such call, TASQ process **537** conventionally measures latency of the network connection for that call by regularly sending a "ping" to the peer gateway associated with this call and measuring round-trip transit time. The TASQ process also intermittently polls the DSP driver to obtain the packet loss statistics and buffer under/over-flows for each such call. TASQ process **537** then interpolates and, over time, filters this data, received from the DSP driver and, in conjunction with its latency determinations for that call, determines a numeric grade of the network connection then carrying that particular VoIP call. If the grade of this connection is less than a predefined threshold, then the network quality will be deemed insufficient to handle this call. The TASQ process issues an instruction to call handler **560** to switch the call from the private data network to the PSTN. Alternatively, if the call is then being carried over the PSTN, TASQ process **537** continues measuring network quality to determine whether the network quality, which may have decreased below the threshold, has later increased to above the threshold necessary to support the call and thus again provide toll bypass and cost savings. To do so, TASQ process **537** regularly sends a "ping" from its gateway to its peer gateway (e.g., gateways **200** and **200'** shown in FIG. **1**) which collectively serve the calling and called locations for that call. If the numeric grade associated with this gateway-to-peer gateway connection (i.e., over which this call could be routed) is sufficient, based on these latency measurements, then network quality will now be deemed sufficient to support a VoIP call. In this instance, TASQ process **537** issues an instruction to the call handler to switch the call back from the PSTN to the private data network. As such, TASQ process **537** instructs the call handler to switch the call back and forth between the PSTN and private data network based on dynamic changes in QoS then available through the private data network to provide maximum use of the private data network consistent with the QoS it then provides. Inasmuch as techniques used to measure jitter, packet loss and latency and determine quality of a network connection based on these measurements, including through interpolation and filtering, are all conventional in the art, we will omit discuss them in any further detail. Moreover, since PSTN circuit-switched connections consistently provide a very high uniform level of quality, there is no need to specifically measure QoS of these connections; it can simply be assumed to be sufficiently high at all times.

DSP driver **519** also detects, through the presence of in-band facsimile or modem tones, whether a VoIP call then being established through a gateway carries facsimile or modem data and, if so, through suitable interaction with gatekeeper process **700**, appropriately changes the compression it uses to handle this data. Since the details of these operations are not relevant to the present invention, they will not be discussed any further.

As such, the reader can now readily appreciate that VoIP packets flow through software **500** in the directions and along the data paths given by thick solid lines **518**, **522**, **531**, **534** and **540** among the VPH process, the DSP driver, the

DSPs, the TCP/IP process, the Ethernet driver, and the LAN and private IP data network.

Idle process **502** operates in a preemptive priority mode but at a lowest execution priority (other than an internal idle process executed within the O/S). Process **502** simply determines the status of the microcontroller, in terms of its current workload, and, to detect code corruption, compares software stored in flash memory **205** (see FIG. **2**) and that then residing and executing from SDRAM **210** and generates an error event should it detect an inconsistency.

LED Driver **566**, contained within section 500_B shown in FIG. **3**B, suitably energizes, under program control, various LED indicators **569** to indicate current status information, as provided by the configuration manager, of the gateway.

TSI (time-slot interchanger) driver **585**, which is the sole component within call switching section **580**, provides a control interface to TDM switch **250** and, by doing so, permits call handler **5** to control operation of the switch in order to establish proper time-slot connections therethrough to connect specific time slots associated with caller and called telephony endpoints.

Call signaling section **590** contains three processes: TIAB process **575**, Q.921 process **572** and Q.931 process **577**; and three drivers: AB bit driver **591**, HDLC (D channel) driver **592** and T1/E1 common driver **574**. As noted above, section **590** generates appropriate telephony signaling information for use by the PSTN or the PBX for properly routing calls, either through the PSTN or over the data network, between the gateway and the PSTN, and between the gateway and the PBX, respectively.

As noted above, T1/E1 communication links can utilize either channel associated signaling (CAS) or common channel signaling (CCS). T1AB process **575** interacts, through AB bit driver **591**, with individual signaling bits A, B provided in CAS and converts the signaling information contained in these bits into a representation usable by the call handler. Process **575**, interacting with this driver, also collectively implements a reverse function of converting signaling information provided the call handler into these individual signaling bits. Process **575** and driver **591** are activated only if the T1/E1 link is operated in the CAS mode.

T1/E1 common driver **574**, which is utilized with both CAS and CCS mode T1/E1 links, interacts with and controls T1/E1 transceiver/framers **260** and **270** (see FIG. **2**) and implements a portion of T1/E1 framing common to both CAS and CCS. This driver detects T1/E1 alarm conditions and sends such detected conditions to call handler process **560**, shown in FIG. **5** (though a linkage between these two processes is not specifically shown). HDLC (D-channel) driver **592** (shown in FIG. **5**), along with Q.921 process **572** and Q.931 process **577**, are all activated only if the T1/E1 links are operated in the CCS mode.

HDLC driver **592** controls operation of a corresponding serial communication circuit (SCC), for use as an HDLC, situated within the microcontroller. This driver is connected, under program control, to a different B- (64 kbits/sec) or D- (16 kbits/sec) channel within the PSTN/PBX T1/E1 link to control data transmission and reception through that particular HDLC from or to a TDM channel on that link. This driver, depending on a direction of data flow through it, either extracts Q.921 messages from information appearing on this TDM slot and applies those messages to Q.921 process **572**, or operates in the reverse direction to apply Q.921 messages generated by Q.921 process **572** for carriage through that TDM time-slot. The HDLC driver is assigned to specific SCCs, acting as HDLCs, under program control as conditions warrant. These HDLCs along with a physical layer T1/E1 interface (not specifically shown in either FIG. **2**) collectively implement, in hardware, layer **1** of OSI network functionality. Under an event-driven software-implemented supervisor, the microcontroller, in view of current resource requests and then available hardware resources, assigns driver **592** to a given SCC, which is then to be used as an HDLC, in order to handle a desired call through the gateway (e.g., call send or receive).

Q.931 process **577** is conventional and encodes outgoing signaling messages from CH process **5** into appropriate Q.931 message form for communicating these messages through D-channel signaling to the PSTN or PBX to control call setup and tear-down. Resulting outgoing Q.931 messages are appropriately encapsulated in a Q.921 information frame, by Q.921 process **572**, for subsequent transport over D-channel signaling to either a local central office switch or the local PBX. The processes collectively operate in a reverse direction to process incoming Q.921 information frames and decode incoming D-channel Q.931 signaling messages into appropriate signaling messages for processing by call handler **560**. Q.931 process **577**, along with Q.921 process **572**, collectively implement, in software, well-known layers **3** and **2**, respectively, of OSI network functionality.

Call processing section **550** contains gatekeeper process **700**, border element process **900** (which includes a counterpart peer border element process, not specifically shown), event server **555**, call handler process **560**. H.323 driver **563** and P.323 process **553**. As discussed above, this section manages the H.323 environment in which the gateway functions; allocates call handling resources to process a call; and routes telephony calls over either the PSTN or data network and, through interaction with VPH process **517** and TASQ process **537**, switches those calls back and forth between the PSTN and the data network, as warranted by QoS then being provided over the data network. Gatekeeper process **700** and border element process **900** utilize configuration information stored in database **508**, as well as write such information into the database. These two processes have been discussed above to the extent necessary for a clear understanding at this level, but will be discussed in considerably greater detail below in conjunction with lower-level block diagrams shown in FIGS. **7**, **9**–**10** and messaging, shown in subsequent diagrams, that flows within a gateway and between peer gateways to implement inter-gateway telephony call processing.

Call handler (CH) process **560** implements all call control functions in the gateway. In particular, the call handler process appropriately routes call between particular trunks between the PBX and either the data network or PSTN—with "trunks" in this context being viewed as a logical entity encompassing communication channels to the PBX, the PSTN or the data network. Through an internal auto-switch manager, the call handler also implements auto-switching functionality in a zone, i.e., switches a telephony call between PSTN and data network connections, in response to dynamic changes in QoS conditions on the data network. The CH also handles signaling protocols related to incoming and outgoing calls.

In particular, CH process **560**, through interaction with both DSP driver **519** and TASQ process **537** as to current QoS of a data network connection, routes a telephony call over either the PSTN or through the data network, through directing an incoming TDM call from the PBX, via the TDM switch, either out to the PSTN or to VPH process **517**. This latter process, as discussed above, directs G.711 packetized telephony information to a DSP, through DSP driver **519**,

which converts this information into suitable G.723 compressed IP packets, which, in turn, are routed, via TCP/IP process **535** and Ethernet driver **533**, onto the LAN connection to the private data network. CH process **560**, in response to instructions from TASQ process **537** resulting from its dynamic determinations of QoS of the private data network, switches a call back and forth between the PSTN to the IP network, consistent with changes in the QoS. Furthermore, CH process **560** implements selective call routing through which that process determines, based on predefined called number information, e.g., predefined called numbers and lists of bypass telephone numbers (BPNs) and exchanges, stored as configuration information within database **508**, whether certain calls, such as emergency calls or local calls, must be routed through the PSTN and not through the private data network, and routes those calls accordingly. CH process **560** also processes, through message passing, incoming and generating outgoing T1/E1 signaling messages, received from either T1AB process **575** or Q.931 process **577**, to establish proper call routes through the PSTN and local PBX, i.e., to perform incoming call routing for those directory numbers which the call handler effectively terminates. CH process **560** also manages T1/E1 channels, and allocates and assigns an individual DSP, via DSP driver **519**, to a corresponding T1/E1 TDM channel for use in initiating a corresponding VoIP call and thereafter for performing voice processing for the duration of that call, after which CH **560** frees that DSP for subsequent re-assignment and use for another such call. CH process **560** also processes T1/E1 alarms, as discussed above, detected by T1/E1 common driver **574**. Furthermore, the CH process controls, as noted above and through TSI driver **585**, operation of the TDM switch within the gateway. CH process **560**, operating in conjunction with H.323 protocol stack **563** through function calls, processes, in accordance with the H.323 standard, incoming H.225.0 call control messages and generates such outgoing H.225.0 messages. Inasmuch as stack **563** is implemented using a library, i.e., this stack is process-less, portal H.323 process (P.323 process) **553** provides a TCP/IP transport layer interface to the stack. Process **553** connects through a socket into TCP/IP process **535** and performs read and write operations into H.323 stack **563** in order to provide H.225.0 messages destined to CH process **560** and transport such messages generated by the CH process over the LAN and, from there, the private data network.

Event server **555** communicates with CH process **560** and collects and stores call events and also implements a server side of the Telnet protocol on a port number other than standard telnet port **23**. A telnet client or custom application executing on a user PC or workstation can communicate with this process and read the stored call events. These call events, which include, e.g., call connect and call disconnect, can be subsequently processed through that PC or workstation to generate, e.g., so-called "call detail recordings" for later use in call accounting and billing or other purposes.

Though, for simplicity, we have shown and described call processing software **550** as containing a single gateway and a single border element, this software can implement multiple different gatekeepers and multiple different border elements, each being a different instance of gatekeeper process **700** and border element process **900**, respectively, depending on those portions of an actual network topology implemented through a single gateway. In addition, for large network topologies, any instance of the gatekeeper and border element can be implemented through an external computing system, such as a personal computer or workstation, that has appropriate network and software interfaces to the remainder of the gateway. Since such interfaces would be conventional and readily apparent to those skilled in art, we will omit all such details thereof.

FIG. **6** depicts table **600** which shows relative execution priorities of processes that constitute call processing software **500**. As can be seen, TCP/IP process **535** and CM process **505** each possesses a highest relative execution priority (value **255**). Assigning such a priority to TCP/IP process **535** minimizes VoIP call latency through the gateway. Assigning an equally high priority to CM process **505** ensures that this process will properly and regularly, through watchdog timer driver **507**, reset the watchdog timer while the gateway is normally operating and thus prevent that timer from inadvertently, due to a busy processing workload, expiring and generating an erroneous catastrophic alarm condition. VPH **517** is assigned a next lower priority level (value **200**) but which is sufficiently high, given that the VPH lies in the data path for VoIP packets, to substantially reduce any latency which these packets might otherwise experience through the gateway. Both gatekeeper process **700** and border element process **900** (along with its peer border element process) are each allocated relative execution priority **150** with various call control and signaling processes, specifically CH process **560**, TIAB process **575**, P.323 process **553**, Q.931 process **577** and Q.921 process **572** all sharing a next lower execution priority value of **100**. Idle process **502** is assigned a lowest execution priority (value **10**), apart from an idle process internal to the O/S, with all other processes utilized within call processing software **500** sharing a higher relative execution priority value (of **50**) than that of Idle process **502**.

a. Gatekeeper process **700**

FIG. **7** depicts a block diagram of the software that implements gatekeeper process **700**. Gatekeeper process **700** implements each of gatekeepers 420_1, 420_2, 420_3, 460_1 and 460_2 depicted in FIG. **4B**.

As depicted in FIG. **7**, gatekeeper process **700** contains user interface **710**, external API (application programming interface) **720**, system management process **730**, administrative domain client manager **740**, endpoint manager process **750**, routing process **760**, system administration process **770**, H.225.0 process **780** and IP process **790**.

Block **710** implements a user interface to the entire gatekeeper for determining and diagnosing problems with the gatekeeper, as well as for user management of the gatekeeper and to obtain stored statistics from the gatekeeper. External API block **720** provides an API interface which can be used to extend functionality of the gatekeeper for its integration into a larger system, such as a call center or automatic call distributor. Routing process **760** implements endpoint routing. In particular, routing process **760** contains internal routing table **765** which specifies routing information, in terms of directory numbers, endpoint aliases and H.323 endpoint identifiers, for all H.323 endpoints in the zone for which that gatekeeper manages. To reduce processing delays, routing process **760** stores, in an internal cache memory (not shown), endpoint addresses to which that process has recently routed calls. System administration process **770** implements various administrative functions, such as bandwidth and zone management, policy and admissions of entities, such as endpoints or gateways, in the zone.

Administrative domain client manager **740** implements appropriate functionality necessary for establishing service relationships between a gatekeeper and a border element in order for the gatekeeper to publish its call routing capability, through that border element (and to its peer border element),

to every other gatekeeper in the same administrative domain. Manager **740** also resolves addresses that are not contained within either an internal routing table, specifically table **765**, residing within a gatekeeper or an external table (as discussed below in conjunction with border element **900** shown in FIG. **9**) associated with that gatekeeper, or in cache memory associated with that gatekeeper.

Endpoint manager **750** manages H.323 endpoints which includes registration and de-registration of endpoints (e.g., gateways and telephony endpoints—both of which are viewed as "endpoints" under the H.323 standard), allocating and de-allocating network bandwidth associated with a call, call routing between telephony endpoints and appropriate endpoint address translation for use by routing process **760**. This translation applies to outgoing calls and entails conversion of a calling directory number into one or more IP address, as needed. The gatekeepers also utilizes an external table (not shown) which provides routing information for all other H.323 endpoints in the same administrative domain as the gatekeeper.

H.225.0 process **780** processes the H.225 protocol and, as such, encodes and decodes H.225.0 messages accordingly outgoing from and incoming to the gatekeeper, respectively, to or from a border element or H.323 endpoint. IP process **790** implements UDP, TCP and IP network layers of the TCP/IP protocol, and interacts, as needed, with all the other processes in the gatekeeper to provide network communications.

Lastly, system management process **730** configures the entire gatekeeper, supervises operation of the gatekeeper, and both gathers operational statistics from and manages faults for the gatekeeper. Illustratively, an SNMP client (not shown), embedded within block **730**, is used to manage and communicate fault information to a requesting process.

We will now discuss the general process interactions shown in FIG. **7** with, where appropriate, the corresponding messages shown in parenthesis.

As shown, system management process **730** communicates with both endpoint manager **750** and administrative domain client manager **740**. Through interaction with the endpoint manager, system management process **730** sets a bandwidth table that endpoint manager **750** utilizes in allocating and de-allocating network bandwidth to calling endpoints (Set Bandwidth), as well as setting different IP addresses to which the gatekeeper will listen for H.323 discovery request and registration request messages. Bandwidth allocation and de-allocation include reserving available network bandwidth for a call, adding bandwidth to a call then in progress and freeing network bandwidth no longer needed for a call. System management process **730** also communicates with administrative domain client process **740**. Through interaction with manager **740**, the system management process sets IP addresses of each border element in the administrative domain. Client manager **740** also communicates with routing process **760**. Administrative domain client manager **740**, through interaction with routing process **760**, can clear all external route entries in the external routing table associated with routing process **760** (Flush Network Router), add external route entries to that table (Add Network Address), change an external address (Update Network Address) or remove an external address from routing process **760** (Delete Network Address).

Endpoint manager **750** bi-directionally communicates with administrative domain client manager **740**. Through interaction with manager **750**, administrative domain client manager **740** notifies (Connect) the former that a specific border element in the administrative domain has connected to another border element external to the domain in order to facilitate call routing therebetween. Also, endpoint manager **750**, by interacting with manager **740**, informs the latter that a new endpoint has just registered (Endpoint Register) with the gatekeeper or an existing endpoint that had so registered has just de-registered itself (Endpoint De-register).

Furthermore, endpoint manager **750** also communicates with routing process **760**. Manager **750**, through interacting with routing process **760**, requests, for a call to be routed (Route Request), a list of destinations for that call from process **760**; adds a zone address to routing table **765** (Add Zone Address); changes a zone address in this table (Update Zone Address); and removes a zone address from this table (Delete Zone Address).

Endpoint manager **750** also receives and sends, through H.225.0 process **780**, standard H.225.0 messages to H.323 elements in the zone managed by gatekeeper **700** and receives such messages from these elements. These messages include those set forth in Table 1 below, some of which will be discussed in detail below in the context of messaging sequences that are used in implementing inter-gateway call routing and associated call handling procedures.

TABLE 1

STANDARD H.225.0 MESSAGES

H.225.0 Message	Purpose
GRQ	Gatekeeper request -- sent by an endpoint to discover gatekeepers to which that endpoint can register
GCF	Gatekeeper confirm -- sent by a gatekeeper, which will register that endpoint, to confirm a GRQ and identify itself to the requesting endpoint
GRJ	Gatekeeper reject -- sent by a gatekeeper, in response to a GRQ, where the gatekeeper discovery request is rejected by a gatekeeper
RRQ	Registration request -- sent by an endpoint to its corresponding gatekeeper in order to register itself with that gatekeeper, serve as a basis for "keep alive" signals and transfer call routing information to that gatekeeper, i.e., what PBX stations and PSTN numbers are serviced by that endpoint
RCF	Registration confirm -- sent by a gatekeeper to a corresponding endpoint to confirm an RRQ
RRJ	Registration reject -- sent by a gatekeeper, in response to an RRQ, when the registration request is rejected
ARQ	Admission Request -- sent by a registered endpoint when it is answering or originating a call. This message permits the gatekeeper to screen the call to determine if it is allowed with respect to, e.g., bandwidth restrictions, security restrictions or other reasons.
ACF	Admission Confirm -- sent by a gatekeeper, in response to an ARQ, to allow an endpoint to complete an attempted call
ARJ	Admission Reject -- sent by a gatekeeper, in response to an ARQ, to prohibit an endpoint from completing the call on the data network
DRQ	Disengage Request -- an indication sent and received by a gatekeeper, and initiated by an endpoint or gatekeeper, that a call then in progress is to be dropped
DCE	Disengage Confirm -- sent by a gatekeeper, in response to a DRQ, to confirm acceptance of a disengage request
DRJ	Disengage Reject -- sent by a gatekeeper, in response to a DRQ, if an endpoint requesting to be disengaged is not registered
URQ	Unregistration request -- sent by an endpoint to de-register itself from its gatekeeper
UCF	Unregistration confirm -- sent, by a gatekeeper to confirm receipt of the URQ
URJ	Unregistration reject -- sent by a gatekeeper to reject a unregistration request
BRQ	Bandwidth Request -- sent by a registered endpoint to request for additional bandwidth for use with an established call, such as, e.g., for use with a facsimile or modem call

TABLE 1-continued

STANDARD H.225.0 MESSAGES

H.225.0 Message	Purpose
BCF	Bandwidth Confirm -- sent by a gatekeeper, in response to a BRQ, and indicates maximum allowed bandwidth requesting endpoint can use for its call
BRJ	Bandwidth Reject -- sent by a gatekeeper, in response to a BRQ, if the gatekeeper is unable to identify the call for which bandwidth is being requested
IRQ	Information Request -- used by a gatekeeper to keep in synchronization with each endpoint in its zone. Each gatekeeper periodically sends this message to each of its registered endpoints to obtain current call status information for that endpoint.
IRR	Information Response -- response from an endpoint to a received IRR message, or an unsolicited status report from an endpoint to a gatekeeper regarding a specific call

User interface **710**, external API **720** and IP block **790** communicate with all the other constituent elements in gatekeeper **700**. However, to simplify the drawings, the links between the former three blocks and the latter elements have all been intentionally omitted from FIG. **7**.

b. Call handler process **560**

FIG. **8** depicts a block diagram of call handler process **560**. In essence, the call handler is responsible for routing calls between trunk groups. For the purposes of the call handler, a trunk group is a logical entity. Separate such trunk groups are associated with the PBX, the PSTN—the latter two being, e.g., physical T1 (or E1) trunks, and the data network (H.323). A trunk group is associated with a physical signaling method, either CCS (pri), CAS or IP and whether that group is used for the PBX, PSTN or IP (H.323).

In particular, call handler process **560** appropriately routes call between particular trunks between the PBX and either the data network or PSTN. Through internal auto-switch manager **810**, the call handler also implements auto-switching functionality in a zone, i.e., switches a telephony call between PSTN and data network connections, in response to dynamic changes in QoS conditions on the data network. The CH also handles signaling protocols related to incoming and outgoing calls.

Call handler process **560** contains auto-switch manager **810**, H.323 manager **820**, CAS manager **830**, PRI manager **840** and call handler manager **850**.

Auto-switch manager **810** manages auto-switching by allocating auto-switch identifiers, i.e., identifying, through the CallIds and Calling and Called flags, which calls can be auto-switched, and handling relationships between active IP calls and circuit-switched calls.

H.323 manager **820** provides an interface between a trunk group and H.323 stack **563** (see FIG. **5**). This manager also converts appropriate messages into call control functions.

CAS manager **830**, shown in FIG. **8**, provides an interface between a trunk group and a physical trunk (T1 or E1) using CAS (channel associated signaling). This manager sends and receives call progress messages to a physical channel. PRI manager **840**, is similar to CAS manager **830**, provides an interface between a trunk group and a physical trunk using CCS (common channel signaling) rather than CAS.

Lastly, call handler manager **850** contains a list of all the trunk groups that have been configured for use by the gateway. When a call request occurs on any of these trunk groups, the call handler manager determines which trunk group is to receive the call. This manager also routes messages, through various managers in call handler **560**, between different trunk groups.

As shown, interactions occur between CAS manager **830** and Call Handler Manager **850**, where the latter routes messages between the CAS manager to and from any of the other managers in call handler **560**; hence using the call handler manager as a "relay station" between any of these other managers and CAS manager **830**. For simplicity, we will only consider call control messages that are passed between CAS manager **830** and call handler manager **850**, even though the latter manager may have passed these messages onward and received responses from any of the other managers prior to issuing its response to manager **830**. The messages pertinent to these interactions are particular to CAS.

First, a CH_SETUP message is received by CAS manager **830** indicating a call request is made on a trunk group. In response, Call Handler manager **850** issues a CALL_SETUPACK message back to CAS manager **830** to inform the latter that manager **850** has accepted the setup message and the call for which setup is requested can be routed. CAS manager also provides a CH_DISC message to manager **850** specifying that particular call can be disconnected. CAS manager **830** also provides a CH_CALLPROC message to manager **850** indicting that an outgoing call is in progress. This particular message contains status information which originates from a switch in the PSTN and indicates that a called directory number has been received by that switch and that switch is attempting to complete the call. CAS manager **830** also provides a CH_ALERTING message to manager **850** to indicate that a call to a called directory number is ringing at that number before that call can be completed. Lastly, CAS manager **830** also provides a CH_CONNECT message to manager **850** specifying that a particular end-to-end voice path has been establish between calling and called numbers.

CAS manager **830** also provides a CH_RESTART message to manager **850** to indicate when a physical trunk has recovered from a loss of synchronization, i.e., a loss of synchronization alarm condition.

Call handler manager **850** can also send a CH_DBUPDATE message to CAS manager **830**, PRI manager **840** and/or H.323 manager **820**, as needed, to indicate that configuration, including routing, information stored in database **508** (see FIG. **5**) has changed.

Manager **850** also receives call control messages from H.323 manager **820**, CAS manager **830** and PRI manager **840**. Here too, manager **850** acts as a "relay station" in relaying call control messages among any of these latter three managers. These messages, which are functionally very similar to those employed with CAS manager **830**, include: a peerRcvSetup message which indicates that a call request is being made, a peerRcvProg message which indicates that a specific call is in progress, a peerRcvAlert message which indicates that alerting is occurring at a called destination, peerRcvConnect which indicates that a specific call has been connected and an end-to-end voice path has been established between calling and called numbers, a peerRcvFacility message which indicates that a facility message has been received, a peerRcvRelease message indicating that a specific call is being ended, and a peerRcgRelComp message indicating that call teardown for a specific call is now complete.

In addition, call handler manager **850** provides an initiate message to auto-switch manager **810** to initiate an auto-switch process in a specific direction and for a particular call specified in the message.

As noted above, call handler **560**, through call handler manager **850**, is also responsible for determining a destina-

tion of all call requests and choosing an appropriate network over which to route each call. In particular, each trunk group has a set of directory numbers associated with it. These numbers can be bypass directory numbers (BPNs); local directory numbers, including just exchange prefixes or area codes, as necessary; or so-called "leaky area directory numbers" (which are toll calls that terminate at the PSTN rather than a PBX and transit through the data network to reduce or eliminate a toll charge for each such call). These numbers are defined during configuration. A trunk group can respond to any type of number for which it is responsible to route calls. When a route is requested to a called directory number, one or more trunk groups are searched to locate a match to that directory number (or just an exchange, if that is configured as a BPN). If a match is found, then that call is routed over that trunk group to that called directory number. However, there are various limitations. Specifically, for calls originating on the data network, only PBX trunk groups are searched for a matching directory number. For calls originating from the PBX, those trunk groups associated with the PSTN are first searched followed by the IP (H.323) trunk groups. Lastly, for calls originating from the PSTN, manager **850** determines whether each such call is an auto-switch call, i.e., a call being auto-switched, and if it is, providing that call to auto-switch manager **810** for subsequent handling. Lastly, if that call is not an auto-switch call, then manager **850** searches the PBX trunk groups for a matching directory number, and then routes the call accordingly.

c. Border element process **900**

FIG. **9** depicts a block diagram of the software that implements border element process **900**. Border element process **900**, through separate instances executing in separate gateways, implements each of border elements **430** and **450** depicted in FIG. **4B**. In addition, a separate instance of this process implements peer border element **430'**.

As depicted in FIG. **9**, border element process **900** contains routing process **910**, system management process **920**, border element manager **930**, Annex G messages process **940**, administrative domain manager **950**, peer border element manager **960**, and IP process **970**.

Routing process **910** contain internal routing table **915** which also specifies routing information, in terms of directory numbers, endpoint alias and H.323 endpoint identification, for all directory numbers serviced by the administrative domain in which the border element resides. This table is used to resolve endpoint address requests incoming from other border elements.

Administrative domain manager **950** manages gatekeepers that have requested service relationships with border element **900**. Manager **950** provides companion (server-side) functionality to administrative domain client process **740** (see FIG. **7**) executing in the gatekeeper. In that regard, manager **950**, shown in FIG. **9**, updates and provides routing services, through routing process **910**, for and between gatekeepers registered in the administrative domain. To route to a called directory number, a calling telephony endpoint requests a gatekeeper to furnish routing information for a called telephony endpoint. If the gatekeeper can located the called endpoint in its internal routing table (e.g., table **7**=in gatekeeper **700** shown in FIG. **7**) or its external routing table, the gatekeeper then returns routing information to the calling endpoint. The external table holds a static database containing routing information for all endpoints in the same zone as the gatekeeper and is modified, by the gatekeeper, with information gathered during registration procedures. The internal database contains such information for all endpoints in the same administrative domain. If, however, that gatekeeper can not locate the called endpoint in either of these two tables, it issues a request to its associated border element to resolve the called endpoint address. This request is processed by administrative domain manager **950** which issues a request to all other border elements that have established a current service relationship with the associated border element. If any of the other border elements, through their internal routing tables, can resolve that endpoint address, that border element will return the requested endpoint address to the associated border element.

Border element manager **930** implements appropriate functionality necessary to establish service relationships with other border elements, i.e., such that border elements registered in the H.323 environment, can pass information amongst themselves. Peer border element manager **960** manages relationships themselves that exist between peer border elements which includes establishing and terminating such relationships. As discussed in detail above, peer border elements are formed, for purposes of increased fault tolerance and redundancy, of a plurality, e.g., a pair, of border elements in the same administrative domain, as illustrated in FIG. **4B**, and which collectively function as a single "logical" border element.

Annex G message process **940** implements a protocol section of the Annex G standard for communications between border elements. IP process **970**, similar to IP process **790** shown in FIG. **7**, implements UDP, TCP and IP network layers of the TCP/IP protocol, and interacts, as needed, with all the other processes in the border element to provide network communications.

We will now discuss the general process interactions shown in FIG. **9** with, where appropriate, the corresponding messages shown in parenthesis.

As shown, system management process **920** communicates with border element manager **930**. Through interacting with manager **930**, management process **920** can add a new administrative domain for use in external call routing (outside of the domain in which border element **900** resides) (Add Service) such that service relationships can be established, via manager **930**, with that new border element, or remove an administrative domain and terminate such relationships, via manager **930**, with the domain being removed. Both border element manager **930** and administrative domain manager **950** communicate with Annex G messages process **940** in order to send and receive Annex G messages to and from other border elements, respectively.

Administrative domain manager **950** communicates with border element manager **930**. The purpose of doing so is to inform manager **930** that a gatekeeper, located within the same domain as border element **900**, has issued a request to the border element to route a call outside this domain (Route Request). Once this occurs, border element manager **930** communicates with routing process **910** (Route Call) in order to obtain, from a border element in another domain, routing information, including a directory number, for routing that call external to the domain.

In addition, administrative domain manager **950** also communicates with peer border element manager **960** to undertake operations to change information stored in a routing table in a peer border element(s). These operations include adding descriptors (Descriptor Add) resulting from a new endpoint that has just registered itself with a gatekeeper, located in the same administrative domain as border element **900**, and which has downloaded all its descriptors to the gatekeeper. The gatekeeper, in turn, supplies those descriptors to the border element to update

routing information stored within corresponding routing tables residing at that border element and its peers. Also, these operations include deleting descriptors (Descriptor Delete) associated with an endpoint as a result of that endpoint having de-registered itself with its gatekeeper. Further, these operations include a border element notifying its peer(s) that a zone has just connected (Zone Connect) to or disconnected from (Zone Disconnect) that border element such that the peer border element can update its routing information.

Administrative domain manager **950** also communicates with routing process **910**. Through this communication, manager **950** instructs routing process **910** to add (Add Network Address) an address into routing table **915**, change an address stored in this table (Update Network Address) or delete an address residing in this table (Delete Network Address).

Also, peer border element manager **960** communicates with routing process **910** in order to update routing information stored in routing table **915** based on information which this manager has received from its peer border element(s). Such updating includes adding an address into the routing table (Add Network Address), changing an address stored in this table (Update Network Address), and removing an address residing in this table (Delete Network Address).

Since IP block **970** communicates with all the other constituent elements in border element **900**, then, to simplify the drawings, the links between block **970** and all the latter elements have all been intentionally omitted from FIG. **9**.

FIG. **10** depicts a state diagram (i.e., for state machine **1000**) for peer border element manager **960** shown in FIG. **9**. To readily distinguish, in this diagram, events from actions, events and actions are preceded by "E" and "A", respectively. For simplicity, this diagram assumes that only one border element is used, in an administrative domain, as a peer border element. Should more than one peer border element exist in any such administrative domain, then state machine **1000** is replicated, as needed, to interact with each different peer border element in that domain.

In the absence of having to exchange information with any peer border element, peer border element manager **960** remains in idle state **1010**. When manager **960** receives a message from administrative domain manager **950** to modify routing information stored in its peer border element, manager **960** then transitions, as represented by line **1013**, its state and attempts to establish, as shown in block **1015**, a client TCP connection with its peer border element, and specifically with an administrative domain manager situated therein. Once this attempt starts, manager **960** transitions, as represented by line **1017**, to startup state **1020**. If the connection could not be established, i.e., a TCP failure event occurred, then manager **960** transitions its state, as represented by line **1021**, to block **1045**. At this point, the manager starts a retry timer and then transitions, as represented by line **1047**, into waiting state **1050**. During this time, either one of two events will happen: either the peer border element will issue a TCP server-side connect request or the retry timer will expire. Once either event occurs, manager **960** transitions, as indicated by line **1053**, to block **1055** through which manager **960** will once again attempt to establish a client TCP connection with the peer border element. Once this attempt starts, manager **960** transitions, as represented by line **1027**, back to startup state **1020**, and so forth.

Alternatively, if a TCP connection is successfully established between the border element and its peer, then peer border element manager **960** transitions, as represented by line **1023**, to block **1025** through which manager **960** triggers an event to cause routing information stored in the border element to be downloaded to its peer border element. Once this download starts, peer border element manager **960** transitions, as represented by line **1027**, to active state **1030**. During this state, information stored in the peer border element is updated, as indicated in block **1035**, in response to update requests issued by administrative domain manager **950**. As represented by lines **1033** and **1037**, manager **960** remains in active state **1030** until such time as all necessary updates have occurred or a TCP failure event occurs. Should the TCP connection fail prior to completion of all the updates, then, as indicated in block **1040**, manager **960** starts the retry timer. Once that occurs, manager **960** transitions its state, as represented by line **1043**, to waiting state **1050**, to attempt re-establishing the TCP connection with the peer border element, and so forth. Alternatively, once all the updates have occurred, manager **960** transitions, as indicated by line **1039**, back to idle state **1010**, and so forth.

D. Inter- and intra-gateway call routing and associated operations

We will now direct our attention to interactions, including messaging, that occurs both between peered gateways as well as within a gateway for routing telephone calls, in accordance with our invention teachings, for two H.323 telephony endpoints, back and forth between the data network and the PSTN, as well as for performing associated operations, such as call teardown, and H.323 endpoint registration and de-registration.

1. Overview

First, however, we will provide overview information, though noted above, which should aid in understanding these interactions in their proper context.

Generally speaking, in an H.323 environment, each gatekeeper, within a domain, passes call control and call routing information, involving endpoints within its zone, to an external border element located within that domain. A gatekeeper, using its own routing tables, can resolve destination addresses for all endpoints within its administrative domain. Hence, if the calling gatekeeper then possesses, within its routing tables, requisite routing information in the form of a descriptor, for a telephony endpoint then being called, that gateway is able to route the call itself and does not need to obtain routing information from a called gatekeeper. If, however, that calling gatekeeper does not possess the routing information needed to route that call, i.e., it can not resolve a destination address for that call since the called endpoint resides in a different administrative domain as the gatekeeper, then that gatekeeper will request that information from its external border element. That element, in turn, will issue a request to a called domain, via an external border element in that domain, for the requisite routing information. That information, accessed from an external border element in a domain containing the called telephony endpoint, will then be passed back to the calling gatekeeper which will then route the call accordingly. Hence, a gatekeeper first attempts to complete a call within its own zone, and thereafter within its administrative domain and finally, in the absence of suitable destination information within that zone or domain, on an inter-domain basis.

Prior to handling any telephony traffic, a gateway, typically shortly after its initialization, must register itself with a gatekeeper as a condition of handling such traffic destined to or from it. This registration procedure will be discussed in further detail below in conjunction with FIGS. **24** and **28**. Once the gateway registers itself, each gatekeeper executing

therein must establish a service relationship with its border element such that the gatekeepers and border elements can interact in order to pass control and routing messages therebetween. The manner through which such relationships is established will be discussed below in conjunction with FIG. 21. Through our inventive peered border elements, a gatekeeper can register with one of the peered border elements, which, in turn, will pass registration messages to its peer in order to effectuate gatekeeper registration across both peered elements. Thereafter, each active telephony endpoint that exists within the same zone as a gatekeeper will register its presence, one at a time, with that particular gatekeeper—in much the same manner as a gateway registers itself with a border element. As each such endpoint registers itself with a gatekeeper, that gatekeeper provides the registration information, in the form of a descriptor, to its border element. That element, in turn, publishes the descriptor to every other gatekeeper in the administrative domain. The process through which information is disseminated by the border element will be discussed below in conjunction with FIG. **22**. With this published information, gatekeepers can route calls to any telephony endpoint within its administrative domain without requesting routing information from its border element. As each gatekeeper registers with a border element, and to that extent that gatekeeper contains stored descriptors, that gatekeeper will share those descriptors with the border element for subsequent publication to all other gatekeepers throughout the domain which, in turn, will update their routing tables with these descriptors. Hence, border elements build administrative domain call routing information by accumulating the call routing capabilities of each and every zone it then handles.

In a converse fashion, within an administrative domain, gateways and gatekeepers can terminate their service relationship with each other, and telephony endpoints and gatekeepers can likewise do so between themselves. This can arise through a failure of an H.323 element which after being detected, by a functioning element, forces all service relationships involving that former element to be severed such that the former element is effectively removed from the domain. Alternatively, an element can also request, as in a case of being taken out of service for maintenance, that all of its established service relationships be terminated. De-registration procedures will be discussed below in conjunction with FIGS. **28** and **29**. Hence, call routing information in a domain stored in any active gatekeeper in that domain dynamically changes as gateways and telephony endpoints register and de-register themselves and is supplied to and disseminated by border elements, that have service relationships with those gatekeepers, throughout the domain.

A call descriptor identifies call routing capabilities for zones and administrative domains. Descriptors have at least one template. A template contains a profile for either one H.323 endpoint or a range of different endpoints. One attribute of the template is a routing information field which indicates whether the endpoint can be contacted directly or must be resolved dynamically. For each endpoint, its template identifies, inter alia, its directory number, alias(es) and its IP address on the private data network. These aliases may include, e.g., H.323-IDs, url-IDs, transport IDs, and/or e-mail-IDs.

In accordance with our invention and as noted above, auto-switching between the private data (IP) network and the PSTN occurs in response to dynamic changes in the quality of a connection over the private data network. Auto-switching begins in a gateway. A gateway, through TASQ process **537** (see FIG. **5**), determines network quality through dynamic measurements of latency, packet loss and error rate (jitter). Should either gateway involved in a call determine that network quality has either increased or decreased to necessitate an auto-switch either to the data network from the PSTN or the opposite, that gateway (hereinafter, for simplicity of reference, the "calling gateway") will initiate an information exchange, using, in accordance with our inventive teachings, certain call-specific embedded, as "nonstandard Data" within specific H.323 message, with its peer gateway (hereinafter, the "called" gateway).

If the call is to transition from the data network to the PSTN, the called gateway will select an available directory number from a pool of directory numbers that has been assigned to it (so-called "pooled directory numbers" or PDN) during its configuration and convey that specific number to the calling gateway. Once the calling gateway receives the particular PDN, it originates a circuit-switched call over its PSTN trunk connection to that PDN. The called gateway, sensing an incoming call on its PDN, will determine whether this number corresponds to the particular PDN on which that gateway is now expecting a call. If it is a different PDN number from that which it is expected, that gateway sends a message to the calling gateway over the network connection and waits for a gateway to claim this call. If this call is on the correct PDN, then the called gateway switches the call, through suitable instructions given to its 4-by-4 TDM switch **250** (see FIG. **5**) to switch the call from its network connection to the now established circuit-switched connection through the PSTN. Once this occurs, the data network connection for this call is torn down by both gateways as if the call were completed. Auto-switching also occurs in reverse, from the PSTN back to the data network, when network quality sufficiently improves. We will discuss below and in conjunction with FIGS. **16–18** the call processing and inter-process and inter-gateway messaging which collectively implement auto-switching.

Furthermore and as noted above, in accordance with our present invention, we embed certain call-specific information within specific H.323 message conveyed between gateways on opposite sides of a call to effectuate auto-switching of that call between the PSTN and the data network. By virtue of this information, the gateways on calling and called sides form the same association for each call routed therebetween and with a common though unique identifier (CallId) used for that call. This identifier distinguishes that call from any other then being handled by either gateway such that the gateways, acting in unison, can switch this particular call between these networks, as needed, without affecting any other calls.

Specifically, we embed, through call independent signaling, a Calling Flag within a conventional "nonstandard Data" field in an H.323 SETUP message, and a Called Flag; a CallId and a selected PDN all within a conventional "nonstandard Data" field in an H.323 CALL PROCEEDING message. In that regard, the contents of the Calling Flag, which are generated by a calling side, contains information, for a given call being established, which indicates whether the call, from a perspective of a calling gateway, can be auto-switched. In response to this SETUP message, the called side generates a CallId number which uniquely identifies that call and then passes that ID back to the calling side, along with the Called Flag and PDN. The Called Flag specifies whether the call, from a perspective of the called gateway, can be auto-switched. The calling side then saves this information for later use in properly auto-switching the

call between the data network and the PSTN, should a need to auto-switch then occur. By virtue of exchanging this information, the calling and called sides form the same association, using a common CallId, for each call routed therebetween so as to readily distinguish that call from any other then being handled by a gateway on either side and indicate to each other that the call can be auto-switched should a need later arise to auto-switch that call. In that regard, only those calls for which both the calling and called gateways have indicated as being capable of being auto-switched are eligible to be auto-switched in response to dynamic QoS changes of a network connection between these gateways. Any call for which either the calling or called gateway has indicated, in the Calling and Called Flags, can not be auto-switched will remain on the PSTN regardless of such QoS changes.

Though we have illustratively described above various call-specific information as being embedded in an H.323 CALL PROCEEDING message and will continue to describe it as such in the ensuing description of FIGS. **13–18**, the same information can alternatively be embedded within an H.323 CONNECT message. This information would be embedded in the latter message in much the same fashion as it would be in the former message but with appropriate changes, readily apparent to those skilled in the art, to corresponding call processing operations shown in these figures and discussed below.

2. Basic VoIP call processing

FIG. **11** depicts highly-simplified operational sequence **1100** for processing, in accordance with our present invention, a VoIP call between two H.323 telephony endpoints.

As shown, to institute a VoIP call, a calling telephony endpoint, that has already registered itself with a gatekeeper, first issues, as represented by line **1105**, an H.225.0 ADMISSION REQUEST (ARQ) message to that gatekeeper. In response to this message, the gatekeeper screens, as indicated by block **1110**, its admission policy to determine whether the call now being attempted is allowed, i.e., whether the calling telephony endpoint has requisite permissions to make the call or whether sufficient network bandwidth is then available to support the call. If the call is permitted and sufficient bandwidth is available, the gatekeeper then responds, as represented by line **1115**, with an H.225.0 ADMISSION CONFIRM (ACF) message; otherwise, the call is rejected with the gatekeeper responding with an H.225.0 ADMISSION REJECT (ARJ) message. The ACF message informs the endpoint that it is permitted to complete its intended call via the network connection. The ARJ message prohibits the endpoint from completing this intended call over the data network.

If the call is permitted, then, as represented by line **1120**, the gatekeeper servicing the calling telephony endpoint and in response to a request from that endpoint will "set up" the call by obtaining appropriate routing information, either from accessing its own routing tables if the call is intra-domain or from a suitable external border element if the call is inter-domain, needed to form a packet connection through the private data network between the calling and called endpoints. Thereafter, as represented by line **1125**, the endpoints will undertake appropriate call processing through which the gatekeeper servicing each endpoint involved in the call will allocate and assign appropriate resources to support the call, such as a DSP, and will commence voice processing for that call and will attempt to establish a packet connection between these endpoints. Once the connection has sufficiently progressed, the called endpoint will alert, as represented by line **1130**, the calling endpoint to its presence. Then, the called endpoint will issue an H.225.0 CONNECT message, as represented by line **1135**, to the calling endpoint to complete the connection. Once the connection has been fully established, as indicated by block **1140**, the VoIP call is active and packetized traffic in the form of packets bearing G.723 compressed digitized speech (or facsimile or analog data) can transit over the packet connection between the calling and called parties for the duration of the call.

At the conclusion of the call, one of the telephony endpoints, here illustratively the calling endpoint, will terminate the connection, effectively going "on-hook". To do so and indicated in Call Termination block **1150**, that endpoint will issue, as represented by line **1155**, an H.225.0 DISENGAGE REQUEST (DRQ) message. This message indicates that the call is to be dropped. Such a message can be issued by either an endpoint, as shown here, or by a gatekeeper. Once this message has been received and accepted by its recipient, here the gatekeeper, the recipient issues an H.255.0 DISENGAGE CONFIRM (DCF) message. In response to transmission and reception of the DCF message, both endpoints issue H.225.0 RELEASE COMPLETE messages to each other, thus terminating the network connection therebetween.

Regardless of whether a call then exists or not between a gatekeeper and each of its telephony endpoints, that gatekeeper will periodically send an H.225.0 INFORMATION REQUEST (IRQ) message to all gateways that are registered to that gatekeeper, hence connecting to those endpoints. The reason for doing so is that gateway-gatekeeper communications utilize UDP, which is not designed to be a reliable protocol. That gateway will respond, as represented by line **1175** with an H.225.0 INFORMATION RESPONSE (IRR) message, containing a list of its then active calls. The gatekeeper will compare this list to a list it locally maintains and correct any discrepancies therebetween; thereby, maintaining in synchronization with its gateway(s). Alternatively, a gatekeeper, in response to certain events, may issue an IRQ message to determine status of a specific call then being handled by a gateway. Such an event may include an H.225.0 GATEKEEPER REQUEST (GRQ) message being issued by a registered gateway or an H.225.0 REGISTRATION REQUEST (RRQ) message. Should such a GRQ or RRQ request message occur, this could indicate an occurrence of a fatal event, such as a system reset or loss of power, or a non-fatal event (such as a gateway issuing an RRQ message for some benign reason). Under these conditions, the gatekeeper does not assume that calls, which were supposedly active, are either in progress or not; hence, it updates its own information based on the responses it then receives from its registered gateway(s).

FIG. **12** depicts basic inter-process operations **1200** for routing a telephone call over a data network connection (PBX-IP-PBX) between two gateways in two different zones.

As shown, assume that a user stationed at a telephony endpoint (not shown in FIG. **12**) connected to PBX **14**, dials "1-732-872-8020" as a called number. This number is sent, as represented by line **1210** over T1 trunk **1213** and as signaling information, to gateway **200** ("calling gateway") that serves this endpoint. This signaling information is conveyed through DTMF (dual-tone multi-frequency) tones, pulses or ISDN D-channel information, as appropriate, over an incoming trunk in T1 trunks **1213**. Gateway **200**, in turn, sends, as represented by line **1215** and over RAS channel **1217**, an H.225.0 ARQ message to a gatekeeper, e.g.,

gatekeeper 420_1, servicing this endpoint. This ARQ message specifies the dialed number as well as an amount of network bandwidth the telephony endpoint desires to use for this call. Based on whether this endpoint is to carry, e.g., voice, or data from a computer modem or a facsimile machine, the bandwidth required will vary accordingly. Generally speaking, a RAS channel is an unreliable channel which is used to convey H.225.0 registration, admissions, bandwidth change, and status messages between two H.323 entities. Upon processing the ARQ message, gatekeeper **4201**, assuming the call is permitted, determines whether it can resolve the called number into a network address. If, as here, it can resolve this number—from its stored descriptors, it returns, as represented by line **1220**, an H.225.0 ACF message containing an IP address of the called gateway. In response to this ACF message, the calling gateway sends, as represented by line **1225** and via H.245 channel **1223**, a Q.931 setup message to the called gateway, here gateway **200'**.

In response to the Q.931 setup message, called gateway **200'** sends, as represented by line **1230**, an H.225.0 ARQ message to a gatekeeper, here illustrative gatekeeper 460_1, that serves the called endpoint. This message requests this gatekeeper to provide authorization to accept an incoming call. If such authorization is granted, gatekeeper 460_1 returns, as represented by line **1235** and over RAS channel **1233**, sends an H.225.0 ACF message back to gateway **200'**. In response to this confirmation message, called gateway **200'** sends, as represented by line **1240**, a Q.931 progress message back to the calling gateway to indicate that equipment at the call destination is in process of setting up the call. In addition, called gateway **200'** initiates a call, as represented by line **1245** over an outgoing trunk in T1 trunks **1247**, to PBX **44** using DTMF, dialing pulses or ISDN, depending on its capability and configuration. Once this call is completed through PBX **44** to the destination endpoint with "answer supervision" being returned, PBX **44** then issues, as represented by line **1250**, an appropriate call answered message back to the called gateway. This gateway, in turn, issues, as represented by line **1255** and via the H.245 channel, a Q.931 connect message to the calling gateway. In response to this message, the calling gateway establishes a voice path, via an incoming trunk in T1 trunks **1213**, through PBX **14**, to the calling endpoint and, as represented by line **1260**, issues a ringing signal on this path to indicate the called endpoint is ringing. Once this occurs and the called party answers, a voice path is established, via a networked data connection, between the calling and called endpoints.

3. Inter-process messaging and interactions

We will now discuss inter-gateway messaging in further detail for various different call scenarios.

In the following scenarios shown in FIGS. **13–18**, both the calling and called sides are assumed to be situated behind a corresponding PBX, consistent with that shown, e.g., with respect to PBXs **14** and **44** shown in FIG. **1**. For ready elucidation, the reader should also simultaneously refer to FIG. **1** throughout the ensuing discussion of FIGS. **13–18**.

In these figures, both the calling and called parties are assumed to be H.323 telephony endpoints, such as telephones **16** and **46**. In each of these figures, each message carried over the data network or PSTN is shown as a heavy solid or heavy dotted line, respectively, with an arrowhead indicating direction of that message. Furthermore, for ease of reference and understanding, primed notation, such as for gatekeeper processes **700** and **700'** shown in FIG. **13**, is used throughout FIGS. **13–25** to indicate identical, though separate, instances of processes executing in corresponding peer gateways **200** and **200'**. Furthermore, to simplify understanding, we will discuss various scenarios where calling gateway **200** initiates auto-switching of a call between the data network and the PSTN and in a reverse direction, even though such action can be initiated by either of the peered gateways handling that call. Since the same operations would occur in the either of these gateways, we will omit discussions of auto-switching being initiated by the called gateway.

a. PBX-IP-PBX call

FIG. **13** depicts typical inter-process control messaging, that would occur both between and within peered gateways **200** and **200'**, for routing a telephone call over data network **30** (PBX-IP-PBX) that connects these gateways.

First, as represented by line **1303**, PBX **14** directs an outgoing call to gateway **200**, i.e., a user stationed at telephony endpoint (here, telephone) **16** dials a called number and that number with appropriate signaling information to the gateway and therein to call handler (CH) **560**. In response, the call handler determines whether adequate network bandwidth exists to support the call and the caller has proper security clearance to make the call. If both conditions are met, the CH allocates an available DSP channel and connects PBX **14**, through TDM switch **250** (and through interaction with TSI driver **585** shown in FIG. **5**), to this DSP channel. Once this connection is established, call handler **560** issues, as represented by line **1306** shown in FIG. **13**, an OPEN VOICE PATH command to voice packet handler (VPH) **517** which, in turn, opens a packetized voice path for this call through the allocated DSP channel. Thereafter, CH **560** issues, as represented by line **1309**, an OPEN CHANNEL command to the DSP driver to open that DSP channel. Next, the CH forms the Calling Flag and provides, as represented by line **1312**, that flag within a SETUP message to H.323 process **563**. Process **563** then embeds the Calling Flag within an H.225.0 admission request message and sends, as represented by line **1315**, that message to gatekeeper **700**. This admission request message, as indicated in FIG. **12**, contains the called directory number.

If the gatekeeper accepts the admission request, gatekeeper **700** determines appropriate routing information such as through interaction with an external border element (not shown) in gateway **200'** and then responds, as represented by line **1318**, with an H.225.0 admission confirm message which contains routing information (e.g., a destination network address) for that call. Once appropriate routing information is obtained for this call, either from within the administrative domain or via another such domain, H.323 process **563** sends, as represented by line **1322**, an H.225.0 SETUP message containing the calling flag to called gateway **200'**. Within the called gateway, H.323 process **563'** processes this setup message, and in doing so, issues, as represented by line **1325**, an H.225.0 ARQ message to gatekeeper process **700'**. If this gatekeeper can accept the call, i.e., sufficient network bandwidth is then available for the gatekeeper to handle this call and the called number has appropriate security clearances to receive the call, gatekeeper **700'** responds, as represented by line **1328**, with an H.225.0 admission confirm message. Gatekeeper **700'** then issues, as represented by line **1331**, a SETUP message containing the calling flag. In response to this message, CH **560'** forms a unique CallId value for this call and saves the calling and called directory numbers for this calls. Thereafter, CH **560'** initiates, as represented by line **1334**, a call to destination PBX **44**. Once this occurs, CH **560'** allocates an available DSP channel and connects, via a TDM connection, PBX **44** to this DSP channel. Once this connec-

tion is established, call handler **560'** issues, as represented by line **1337**, an OPEN VOICE PATH command to VPH **517'** which, in turn, opens a packetized voice path for this call through the allocated DSP channel. Thereafter, CH **560** issues, as represented by line **1340**, an OPEN CHANNEL command to DSP driver **519'** to open that DSP channel. Once this channel is open, CH **560'** instructs VPH **517'**, through issuance, as represented by line **1343**, of a START VOICE PROCESSING message, to commence voice processing over this channel.

Once voice processing has commenced at gateway **200'**, CH **560'** selects one of its available pooled directory numbers and forms the Called Flag for this call. Thereafter, CH **560'** issues, as represented by line **1347**, to H.323 process **563'**, an H.225.0 CALL PROCEEDING message which contains the Called Flag, the selected PDN and the CallId for this call. H.363 process **563'** then transmits, as represented by line **1350**, this message to calling gateway **200**. This message will be received by H.323 process **563** which, in turn, a represented by line **1353**, will pass this message to CH **560**. This CH then saves the Called Flag, PDN and CallId for this call for later use during auto-switching.

Once this information is saved, CH **560** then issues, as represented by line **1356**, a START VOICE PROCESSING message to VPH **517**, to commence voice processing over the DSP channel in this gateway allocated to this call. After H.323 process **563'** in gateway **200'** issues the CALL PROCEEDING message, CH **560'** issues, as represented by line **1360**, and H.225.0 CONNECT message to H.323 process **563'**. This process then transmits, as represented by line **1365**, this connect message to calling gateway **200**. In response to receipt of this message by H.323 process **563**, this process passes the H.225.0 CONNECT message to CH **560** to complete the connection over the data network between the calling and called parties, after which packetized voice traffic for this call transits over this connection.

b. PBX-PSTN-PBX call with CONNECT message delivered to calling side

FIG. **14** depicts typical inter-process control messaging, that would occur both between and within peered gateways **200** and **200'**, for routing a telephone call over a PSTN connection (PBX-PSTN-PBX) between these gateways.

First, as represented by line **1403**, PBX **14** directs an outgoing call to gateway **200**, i.e., a user stationed at telephone 16 dials a called number and that number is passed with appropriate signaling information to the gateway and therein to call handler (CH) **560**. In response, the call handler determines whether adequate network bandwidth exists to support the call and the caller has proper security clearance to make the call. If adequate bandwidth does not exist, e.g., the network is then too congested to fully support the call, but the caller has appropriate permissions to make the call, then CH **560** will route the call over the PSTN.

Generally speaking, through the remainder of this scenario, the calling and called gateways exchange appropriate H.323 call signaling information such that should network conditions later warrant auto-switching the call from the PSTN to the data network, both sides will possess sufficient information to do so. This information is exchanged using a call independent signaling procedure as defined in the H.323 standard to convey such signaling information through H.323 SETUP, CALL PROCEEDING and RELEASE COMPLETE messages. In essence, the SETUP message is sent to establish a PSTN connection to a called directory number with call information, specifically a CallId, then being sent using in-band DTMF signaling over the PSTN connection, from the calling to the called gateway, that transports the call. The RELEASE COMPLETE message contains an acknowledgement field that informs the calling gateway that the called gateway has received the in-band signaled call information and the CallId matches that originally sent to the calling gateway and has now been associated with the current PSTN call. Hence, the acknowledgement signifies that both the calling and called gateways possess and have processed requisite call information to later auto-switch this call to the data network.

Specifically and returning to FIG. **14**, once the call handler determines that the call is to be routed over the PSTN, CH **560** forms the Calling Flag for this call and then embeds this flag within a SETUP message. This SETUP message is provided, as represented by line **1406**, to H.323 process **563** which, in turn, generates an H.225.0 admission request message containing the called number and then passes, as represented by line **1409**, that message to gatekeeper **700**. If the gatekeeper accepts the admission request, it issues, as represented by line **1412**, an H.225.0 admission confirm message to H.323 process **563**. In response, H.323 process **563** sends, as represented by line **1415**, an H.225.0 SETUP message containing the calling flag, over the data network, to called gateway **200'**. Within the called gateway, H.323 process **563'** processes this setup message, and in doing so, issues, as represented by line **1418**, an H.225.0 ARQ message to gatekeeper process **700'**. If this gatekeeper can accept the call, i.e., a called endpoint has appropriate security clearances to receive the call, gatekeeper **700'** responds, as represented by line **1421**, with an H.225.0 admission confirm message to H.323 process **563'**. In response to the admission confirm message, H.323 process **563'** passes, as represented by line **1424**, the SETUP message it received containing the Calling Flag to CH **560'**.

By virtue of receiving the SETUP message, CH **560'** establishes a CallId for this particular call and selects an available PDN for possible later use in auto-switching this call and saves the calling and called directory numbers for this call. Thereafter, this CH issues, as represented by line **1427**, to H.323 process **563'** a CALL PROCEEDING message containing the Called Flag, the selected PDN and the CallId. H.323 process **563'** sends, as represented by line **1430**, an H.225.0 CALL PROCEEDING message containing the Called Flag, the PDN and the CallId, over the data network, to calling gateway **200**. Within the calling gateway, H.323 process **563** applies, as represented by line **1433**, the CALL PROCEEDING message to CH **560**. This CH saves the call information it just received in this message for subsequent use during auto-switching. Thereafter, CH **560** issues, as represented by line **1436**, a conventional Q.931 SETUP message to the PSTN containing the called directory number in order to establish a circuit-switched PSTN connection to this directory number. In response to the Q.931 SETUP message and to signify that the call is being established through the PSTN, a local central office switch that serves the calling gateway responds, as represented by line **1440**, with a Q.931 CALL PROCEEDING message to the calling gateway and specifically to CH **560** therein. This message specifies which T1 channel on an incoming trunk (to the PSTN) will carry this call. In addition, the PSTN issues, as represented by line **1443**, a Q.931 SETUP message, to called gateway **200'**, containing the called directory number and an identification of a T1 channel on an outgoing trunk (to the called gateway) which carries this call. Since this call is uni-directional, i.e., from the calling to the called gateway, only one T1 channel is needed at each end.

In response to receiving the Q.931 SETUP message, CH **560'** establishes, as represented by line **1446**, a call to local

PBX **44**. In addition, CH **560'** allocates an available DSP channel and connects, via a TDM connection, the T1 channel (for receive only) from PBX **44** to this DSP channel. Once this connection is established, call handler **560'** issues, as represented by line **1452**, an OPEN CHANNEL command to DSP driver **519'**. Subsequently, after a called party picks up the telephone with suitable signaling (e.g., answer supervision) being returned, as represented by line **1455**, CH **560'** issues, as represented by line **1458**, a Q.931 CONNECT message to the PSTN. The PSTN routes, as represented by line **1460**, this message to the calling gateway. In response to this message, CH **560** issues, as represented by line **1463**, a SEND message, containing the CallId for this call, to DSP driver **519**. This driver converts this message into in-band DTMF signaling and sends this message, as a CALLID message, over the PSTN connection to called gateway **200'**. Upon receipt of this CALLID message, DSP driver **519'** extracts the in-band signaled CallId from this message and issues, as represented by line **1468**, a RECEIVED message containing the CallId to CH **560'**. In response to the RECEIVED message, CH **560'** disconnects the PSTN channel from the DSP channel (receive side—since that is the only side that has been utilized here) and connects the former channel to the receive side of the PSTN channel. Once this occurs, the called gateway has completed its PSTN connection. Hence, CH **563'** issues, as represented by line **1476**, a CLOSE CHANNEL message to DSP driver **519'** to close this DSP channel then in use. CH **560'** also frees this DSP channel for subsequent re-allocation and re-use. In addition, CH **560'** also issues to H.323 process **563'**, as represented by line **1480**, a RELEASE COMPLETE message containing an acknowledgement. This acknowledgement signifies that the CallId was properly received by the called gateway and that this gateway correctly associated it with the PSTN call. In response to this message, H.323 process **563'** issues, as represented by line **1495**, an H.225.0 disengage request message to gatekeeper **700'** to drop the call as far as the gatekeeper is concerned. Once this gatekeeper has effectively removed ("dropped") this call, the gatekeeper issues, as represented by line **1498**, an H.225.0 disengage confirm message back to H.323 process **563'**. In addition, H.323 process **563'** also issues, as represented by line **1483**, the RELEASE COMPLETE message through the PSTN connection to the calling gateway. In response to receipt of this message, H.323 process **563** issues, as represented by line **1486**, an H.225.0 disengage request message to gatekeeper **700** to drop the call as far as the gatekeeper is concerned. Once this gatekeeper has effectively dropped this call, the gatekeeper issues, as represented by line **1489**, an H.225.0 disengage confirm message back to H.323 process **563**, which, in turn, issues, as represented by line **1492**, an H.323 RELEASE COMPLETE message to CH **560**.

c. PBX-IP-PBX call but without connect message delivered to calling side

FIG. **15** depicts typical inter-process control messaging, that would occur both between and within peered gateways **200** and **200'**, for routing a telephone call over the data network, but in the absence of a CONNECT message being delivered to a calling side. This messaging is quite similar to that shown in FIG. **14** but for the process through which the CallId is communicated between the two gateways. Here, however, rather than sending the CallId once, it is sent continuously once a PSTN connection is established until such time as reception of the CallId is acknowledged by the called gateway through an acknowledgment provided in an H.323 RELEASE COMPLETE message issued by that gateway.

First, as represented by line **1501**, PBX **14** directs an outgoing call to gateway **200**, i.e., a user stationed at telephone **16** dials a called number and that number is passed with appropriate signaling information to the gateway and therein to CH **560**. In response, the call handler determines whether adequate network bandwidth exists to support the call and the caller has proper security clearance to make the call. If, as is the case with the scenario shown in FIG. **14** and discussed above, adequate bandwidth does not exist or the network is then too congested to support the call, but the caller has appropriate permissions to make the call, then CH **560** will route the call over the PSTN.

In the same fashion discussed above in conjunction with FIG. **14**, through the remainder of this present scenario, calling and called gateways **200** and **200'** exchange appropriate H.323 call signaling information, using a call independent signaling procedure provided in the H.323 standard, for subsequent use during auto-switching. This information is exchanged through H.323 SETUP, CALL PROCEEDING and RELEASE COMPLETE messages.

With reference to the particular scenario shown in FIG. **15**, once the call handler determines that the call is to be routed over the PSTN, CH **560** forms the Calling Flag for this call and then embeds this flag within a SETUP message. This SETUP message is provided, as represented by line **1503**, to H.323 process **563** which, in turn, generates an H.225.0 admission request message containing the called number and then passes, as represented by line **1505**, that message to gatekeeper **700**. If the gatekeeper accepts the admission request, it issues, as represented by line **1507**, an H.225.0 admission confirm message to H.323 process **563**. In response, H.323 process **563** sends, as represented by line **1510**, an H.225.0 SETUP message containing the calling flag, over the data network, to called gateway **200'**. Within the called gateway, H.323 process **563'** processes this setup message, and in doing so, issues, as represented by line **1512**, an H.225.0 ARQ message to gatekeeper process **700'**. If this gatekeeper can accept the call, i.e., a called endpoint has appropriate security clearances to receive the call, gatekeeper **700'** responds, as represented by line **1514**, with an H.225.0 admission confirm message to H.323 process **563'**. In response to the admission confirm message, H.323 process **563'** passes, as represented by line **1516**, the SETUP message it received containing the Calling Flag to CH **560'**.

By virtue of receiving the SETUP message, CH **560'** establishes a CallId for this particular call and selects an available PDN for possible later use in auto-switching this call and saves the calling and called directory numbers for this call. Thereafter, this CH issues, as represented by line **1518**, to H.323 process **563'** a CALL PROCEEDING message containing the Called Flag, the selected PDN and the CallId. H.323 process **563'** sends, as represented by line **1520**, an H.225.0 CALL PROCEEDING message containing the Called Flag, the PDN and the CallId, over the data network, to calling gateway **200**. Within the calling gateway, H.323 process **563** applies, as represented by line **1522**, the CALL PROCEEDING message to CH **560**. This CH saves the call information it just received in this message for subsequent use during auto-switching.

Thereafter, CH **560** seizes an available PSTN channel (i.e., goes "off-hook") and sends, as represented by line **1524**, an appropriate signaling message to the PSTN to dial the called directory number. The PSTN, in turn, sends, as represented by line **1526**, an appropriate signaling message to the called gateway signifying the presence of an incoming call to the called directory number. In response to this message, CH **560'** establishes, as represented by line **1528**,

a PSTN call through local PBX **44** to the called number. CH **560** also locates a free DSP channel and connects the PSTN channel to that DSP channel, though only to a receive side. In addition, CH **560'** allocates an available DSP channel and connects, via a TDM connection, the T1 channel (for receive only) from PBX **44** to this DSP channel. Once this connection is established, call handler **560'** issues, as represented by line **1532**, an OPEN CHANNEL command to DSP driver **519'**.

At approximately the same time as the called gateway is opening its PSTN channel, the calling gateway, repeatedly sends, as represented by, e.g., lines **1534**, **1536**, **1538**, and **1540**, SEND messages each containing the CallId preceded by a "#" sign. Each such SEND message is received by DSP driver **519** which, in turn, converts the "#CALLID" information into a DTMF signaling message and then transmits this DTMF message in-band to the PSTN. For simplicity if illustration, only one of the latter instances of this message, i.e., that represented by line **1452**, is shown.

Only four successive SEND message are shown, but as many as needed, each with its concomitant DTMF signaled "#CALLID" information will be sent until the CallId is received by the called gateway. Eventually, during the course of these SEND messages and after a called party picks up the telephone with suitable signaling (e.g., answer supervision) being returned, as represented by line **1533**, CH **560'** issues, as represented by line **1537**, a Call Answered message to the PSTN. Once a PSTN channel is established between the called and calling gateways, eventually ones of the DTMF signaled messages, here represented by line **1542**, will be received by called gateway **200'**. In response to receiving this CallId message, DSP driver **519'** converts the DTMF-signaled message to a RECEIVED message containing #CallId and sends, as represented by line **1544**, the latter message to CH **560'**. This call handler then acknowledges receipt of the CallId by issuing, as represented by line **1546**, a RELEASE COMPLETE message containing an acknowledgement to H.323 process **563'**. This acknowledgement signifies that the CallId was properly received by the called gateway, matches the CallId originally established for this call and that this gateway correctly associated this CallId with this PSTN call. In response to this message, H.323 process **563'** issues, as represented by line **1560**, an H.225.0 disengage request message to gatekeeper **700'** to drop the PSTN call as far as the gatekeeper is concerned. Once this gatekeeper has dropped this call, the gatekeeper issues, as represented by line **1562**, an H.225.0 disengage confirm message back to H.323 process **563'**. In addition, H.323 process **563'** also issues, as represented by line **1550**, the RELEASE COMPLETE message through the PSTN connection to the calling gateway. In response to receipt of this message, H.323 process **563** issues, as represented by line **1552**, an H.225.0 disengage request message to gatekeeper **700** to drop the call as far as that gatekeeper is concerned. Once this gatekeeper has dropped this call, gatekeeper **700** issues, as represented by line **1556**, an H.225.0 disengage confirm message back to H.323 process **563**. Then, process **563** forwards, as represented by line **1556**, the RELEASE COMPLETE message, containing the acknowledgement, it received from the called gateway to CH **560**.

At this point, the calling gateway sends an "*" to the called gateway in response to which both gateways connect the PSTN channel to the calling and called parties.

Specifically, after CH **560** receives the RELEASE COMPLETE message, the call handler disconnects a transmit side of the PSTN channel from the DSP channel and connects the transmit side of the former channel to the PBX channel. In addition, CH **560'** issues, as represented by line **1558**, a SEND message containing "*" to DSP driver **519**. In response to this SEND(*) message, DSP driver **519'** converts the message into DTMF in-band signaling and sends the message using such in-band signaling to called gateway **200'**. Thereafter, CH **560** frees the DSP channel it had just used. and issues, as represented by line **1570**, a CLOSE CHANNEL message to DSP driver **519** to close this DSP channel such that it can be re-allocated and used again later.

In response to receiving this DTMF-signaled message containing an "*", DSP driver **519'** converts the DTMF-signaled message to a RECEIVED message containing "*" and sends, as represented by line **1568**, the latter message to CH **560'**. In response, CH **560'** disconnects the PSTN channel from the receive side of the DSP channel and connects the former channel to a receive side for the PBX channel. Thereafter, the CH **560'** frees the DSP channel it had just used and issues, as represented by line **1572**, a CLOSE CHANNEL message to DSP driver **519'** to close this DSP channel such that it can re-allocated and used again later.

d. IP to PSTN switchover using pooled directory number

FIG. **16** depicts typical inter-process control messaging, that would occur both between and within peered gateways **200** and **200'**, for switching a telephone call from being routed over a data network connection that spans these gateways to a PSTN connection between these two gateways, and specifically where the latter connection was established through use of a pooled directory number.

As shown, assume that QoS, as determined by TASQ process **537** (see FIG. **5** and discussed above) executing in calling gateway **200**, of a data network connection then carrying the call decreased below an acceptable level (i.e., where the numeric grade of the QoS decreased below its predefined threshold, as previously discussed). Accordingly as shown in FIG. **16**, VPH **517** issues, as represented by line **1601**, a SWITCH CHANNEL message and specifically such a message specifying a switch for that call to the PSTN. CH **560**, if it saved a PDN during setup of this call, then allocates a free DSP channel, should one then be available. Once having done this, CH **560** then issues, as represented by line **1604**, a conventional Q.931 SETUP message, containing the PDN as the called number, to the PSTN to establish a circuit-switched call to that number. The local central office in the PSTN which serves gateway **200** then issues, as represented by line **1607**, a conventional Q.931 CALL PROCEEDING message with an identification of a T1 channel in an incoming trunk (from gateway **200**) that will carry that call to the PDN. The local central office serving called gateway **200'** issues, as represented by line **1610**, a Q.931 setup message, to this gateway, specifying the called PDN as well as a T1 channel on an outgoing trunk (to the gateway) on which this call will appear.

CH **560'**, as a result of receiving the Q.931 SETUP message, then allocates a free DSP channel, should one then be available and connects, via the TDM switch in the called gateway, the PSTN channel for this call-to that DSP channel. Once this occurs, CH **560'** issues, as represented by line **1616**, an OPEN CHANNEL message to DSP driver **519'** to open this just allocated DSP channel. Thereafter, CH **560'** issues, as represented by line **1619**, a conventional Q.931 CONNECT message to the PSTN to indicate that the called endpoint is connected to the PSTN channel. In response, the PSTN issues, as represented by line **1622**, a Q.931 CONNECT message to calling gateway **200**. Within this gateway and in response to receipt of this CONNECT message, CH

560 issues, as represented by line 1625, a SEND message, containing the CallId of the present call being switched to the PSTN, to DSP driver 519. This message causes the driver to send, as represented by line 1628, the CallId of this call to called gateway 200' using in-band DTMF signaling.

Upon receipt of the CallId information, DSP driver 519' issues, as represented by line 1631, a RECEIVED message containing the CallId. This message is sent to CH 560' which, in turn, disconnects the PSTN channel from the DSP and connects that channel to the PBX. Thereafter, CH 560' issues, as represented by line 1643, a CLOSE CHANNEL (DTMF) message to close the DSP channel used to receive and process DTMF signaling. Next, CH 560' also issues, as represented by line 1647, a CLOSE VOICE PATH message to VPH 517'. This message causes the VPH to close the voice path previously established in gateway 200' over this DSP channel. CH 560' then issues, as represented by line 1650, a CLOSE CHANNEL (VoIP) message to DSP driver 519' which causes the driver to free the two DSP channels, one used for DTMF and the other then used to process the VoIP call.

Also, after DSP driver 519 has sent the CallId information using in-band signaling, CH 560 located in calling gateway 200, disconnects the PSTN channel from the DSP and connects it to the PBX. Thereafter, CH 560 issues, as represented by line 1634, a CLOSE CHANNEL (DTMF) message to close the DSP channel used to receive and process DTMF signaling. Next, CH 560 also issues, as represented by line 1637, a CLOSE VOICE PATH message to VPH 517. This message causes the VPH to close the voice path previously established in gateway 200 over this DSP channel. CH 560 then issues, as represented by line 1640, a CLOSE CHANNEL (VoIP) message to DSP driver 519 which causes the driver to free the two DSP channels, one used for DTMF and the other then used to process the VoIP call.

Once these operations have occurred, CH 560 issues, as represented by line 1653, an H.225.0 RELEASE COMPLETE message to H.323 process 563. H.323 process 563, upon receiving this message, issues, as represented by line 1659, a disengage request message to gatekeeper 700. The gatekeeper having extinguished its end of the data call issues, as represented by line 1656, a disengage confirm message to H.323 process 563. In response to this message, H.323 process 563 sends, as represented by line 1662, the RELEASE COMPLETE message over the PSTN connection, to called gateway 200'. H.323 process 563', upon receiving this message, issues, as represented by line 1665, a disengage request message to gatekeeper 700'. Gatekeeper 700' having extinguished its end the data call then issues, as represented by line 1668, a disengage confirm message to H.323 process 563' which, in turn, issues, as represented by line 1672, a RELEASE COMPLETE message to CH 560'.

e. IP to PSTN switchover using called directory number

FIG. 17 depicts typical inter-process control messaging, that would occur both between and within peered gateways 200 and 200', for switching a telephone call from being routed over a data network connection that spans these gateways to a PSTN connection between these two gateways, and specifically where the latter connection was established through use of a called directory number, rather than as shown in FIG. 16 with a PDN. As can be seen by comparing FIGS. 16 and 17, the overall control scenario shown in both figures is quite similar; however, the scenario depicted in FIG. 17 occurs where the called side has not previously delivered a PDN for use in auto-switching that call to the PSTN.

As shown, assume that QoS, again as determined by TASQ process 537 (see FIG. 5 and discussed above) executing in calling gateway 200, of a data network connection then carrying the call decreased below an acceptable level. Accordingly as shown in FIG. 17, VPH 517 issues, as represented by line 1701, a SWITCH CHANNEL message and specifically such a message specifying a switch for that call to the PSTN. If a PDN has not been delivered for this call, then CH 560 will determine, from its routing information, if a calling directory number has been delivered for this call. If CH 560 possesses this information, then CH 560 allocates a free DSP channel, should one then be available. Once having done this, CH 560 then accesses from its routing information the called number associated with this calling number. Thereafter, CH 560, issues, as represented by line 1704, a conventional Q.931 SETUP message containing the original called number to establish a circuit-switched call to that number. The local central office in the PSTN which serves gateway 200 then issues, as represented by line 1707, a conventional Q.931 CALL PROCEEDING message with an identification of a T1 channel in an incoming trunk (from gateway 200) that will carry that call to the PDN. The local central office serving called gateway 200' issues, as represented by line 1710, a Q.931 setup message, to this gateway, specifying the original called directory number as well as a T1 channel on an outgoing trunk (to the gateway) on which this call will appear.

CH 560', as a result of receiving the Q.931 SETUP message, then allocates a free DSP channel, should one then be available and connects, via the TDM switch in the called gateway, the PSTN channel for this call to that DSP channel. Once this occurs, CH 560' issues, as represented by line 1718, an OPEN CHANNEL message to DSP driver 519' to open this just allocated DSP channel. Thereafter, CH 560' issues, as represented by line 1722, a conventional Q.931 CONNECT message to the PSTN to indicate that the called endpoint is connected to the PSTN channel. In response, the PSTN issues, as represented by line 1725, a Q.931 CONNECT message to calling gateway 200. Within this gateway and in response to receipt of this CONNECT message, CH 560 issues, as represented by line 1728, a SEND message, containing the CallId of the present call being switched to the PSTN, to DSP driver 519. This message causes the driver to send, as represented by line 1732 and over the PSTN channel, the CallId of this call to the called gateway using in-band DTMF signaling.

Upon receipt of the CallId information, DSP driver 519' issues, as represented by line 1735, a RECEIVED message containing the CallId. This message is sent to CH 560' which, in turn, disconnects the PSTN channel from the DSP and connects that channel to the PBX. Thereafter, CH 560' issues, as represented by line 1747, a CLOSE CHANNEL (DTMF) message to close the DSP channel used to receive and process DTMF signaling. Next, CH 560' also issues, as represented by line 1750, a CLOSE VOICE PATH message to VPH 517'. This message causes the VPH to close the voice path previously established in gateway 200' over this DSP channel. CH 560' then issues, as represented by line 1753, a CLOSE CHANNEL (VoIP) message to DSP driver 519' which causes the driver to free the two DSP channels, one used for DTMF and the other then used to process the VoIP call.

Also, after DSP driver 519 has sent the CallId information using in-band signaling, CH 560 located in calling gateway 200, disconnects the PSTN channel from the DSP and connects it to the PBX. Thereafter, CH 560 issues, as

represented by line **1738**, a CLOSE CHANNEL (DTMF) message to close the DSP channel used to receive and process DTMF signaling. Next, CH **560** also issues, as represented by line **1742**, a CLOSE VOICE PATH message to VPH **517**. This message causes the VPH to close the voice path previously established in gateway **200** over this DSP channel. CH **560** then issues, as represented by line **1745**, a CLOSE CHANNEL (VoIP) message to DSP driver **519** which causes the driver to free the two DSP channels, one used for DTMF and the other then used to process the VoIP call.

Once these operations have occurred, CH **560** issues, as represented by line **1756**, an H.225.0 RELEASE COMPLETE message to H.323 process **563**. H.323 process **563**, upon receiving this message, issues, as represented by line **1760**, a disengage request message to gatekeeper **700**. The gatekeeper having extinguished its end of the data call issues, as represented by line **1763**, a disengage confirm message to H.323 process **563**. In response to this message, H.323 process **563** sends, as represented by line **1765**, the RELEASE COMPLETE message over the PSTN connection, to called gateway **200'**. H.323 process **563'**, upon receiving this message, issues, as represented by line **1768**, a disengage request message to gatekeeper **700'**. Gatekeeper **700'** having extinguished its end the data call then issues, as represented by line **1772**, a disengage confirm message to H.323 process **563'** which, in turn, issues, as represented by line **1775**, a RELEASE COMPLETE message to CH **560'**.

f. PSTN to IP switchover

FIG. **18** depicts typical inter-process control messaging, that would occur both between and within peered gateways **200** and **200'**, for switching a telephone call from being routed over a PSTN connection that spans these gateways to a data network connection between these two gateways. This scenario constitutes a "reverse" auto-switching situation inasmuch as the call is being switched from the PSTN connection to the data network and not the other way. Here, call information is carried within an H.323 SETUP message to indicate to a far end gateway that this call is not originating for a new end-to-end connection but rather a switchover of an existing call is to occur. The CallId is used to properly match the PSTN and data network calls. Auto-switching in this direction does not utilize a PDN or an ability of the PSTN to deliver a calling directory number to the called end. Moreover, the manner in which the call was switched from the PSTN to the data network, i.e., through use of, e.g., a PDN or automatic number identification (ANI), is immaterial for auto-switching that occurs from the data network back to the PSTN.

As shown, assume that QoS, again as determined by TASQ process **537** (see FIG. **5** and discussed above) executing in calling gateway **200**, of a data network connection between the calling and called gateways increases above an acceptable level. Accordingly as shown in FIG. **18**, VPH **517** issues, as represented by line **1801**, a SWITCH CHANNEL message and specifically such a message specifying a switch for that call to the data network. In response, CH **560** allocates an available DSP channel and also issues, as represented by line **1804**, an OPEN VOICE PATH message to VPH **517**. Thereafter, CH **560** issues a SETUP message containing both a CallId for this call (which was previously generated during the original routing of this call, such as over the data network before that call was switched to the PSTN and is now being switched back to the data network) as well as an instruction to switch this call ("switching instruction") to the data network.

This SETUP message is provided, as represented by line **1807** to H.323 process **563** which, in turn, generates an H.225.0 admission request message containing the CallId and the switching instruction, and then passes, as represented by line **1810**, that message to gatekeeper **700**. If the gatekeeper accepts the admission request—i.e., by virtue of, e.g., the calling endpoint having permission to utilize the data network and sufficient network bandwidth then being available to support the call, gatekeeper **700** issues, as represented by line **1814**, an H.225.0 admission confirm message to H.323 process **563**. In response, H.323 process **563** sends, as represented by line **1820**, an H.225.0 SETUP message containing the CallId and the switching instruction, over the data network, to called gateway **200'**. Within the called gateway, H.323 process **563'** processes this setup message, and in doing so, issues, as represented by line **1823**, an H.225.0 ARQ message to gatekeeper process **700'**. If this gatekeeper can accept the call, i.e., a called endpoint has appropriate security clearances to receive the call and sufficient bandwidth exists at this end to handle the call, gatekeeper **700'** responds, as represented by line **1826**, with an H.225.0 admission confirm message to H.323 process **563'**. In response to the admission confirm message, H.323 process **563'** passes, as represented by line **1829**, the SETUP message it received to CH **560'**. This call handler then allocates a free DSP channel for voice processing that will occur with this call. Once this DSP channel is assigned, CH **560'** issues, as represented by line **1832**, an OPEN VOICE PATH command to VPH **517'** which, in turn, opens a packetized voice path for this call through the allocated DSP channel. Thereafter, CH **560** issues, as represented by line **1835**, a CONNECT message to H.323 process **563'**. This process sends, as represented by line **1840**, this CONNECT message to the calling gateway. Also, once CH **560'** issues the CONNECT message, this CH then disconnects the PBX channel from the PSTN channel, which previously carried the call, and connects the former channel to the allocated DSP channel. Once this connection has been made with the called gateway, CH **560'** issues, as represented by line **1853**, an OPEN CHANNEL message to DSP driver **519'** to open this DSP channel. CH **560'** also issues, as represented by line **1856**, a START VOICE PROCESSING message to VPH **517'** instructing it to commence voice processing of signals occurring over this DSP channel for a duration of this data network call. Similarly, within calling gateway **200**, in response to the CONNECT message forwarded, as represented by line **1843**, to CH **560**, this call handler disconnects the PBX channel from the PSTN channel, which previously carried the call, and connects the former channel to the allocated DSP channel. Thereafter, CH **560** issues, as represented by line **1846**, an OPEN CHANNEL message to DSP driver **519** to open this DSP channel, followed by a START VOICE PROCESSING message, as represented by line **1850**, to VPH **517** to instruct it to commence voice processing of signals occurring over the DSP channel for the duration of this data network call. CH **560'** also issues, as represented by line **1856**, a START VOICE PROCESSING message to VPH **517'** instructing it to commence voice processing of signals occurring over its DSP channel for this data network call and for a duration of this call. Thereafter, CH **560** issues, as represented by line **1860**, a DISCONNECT message over the PSTN channel to the data network to disconnect the PSTN connection for this call. The PSTN then issues, as represented by line **1863**, a DISCONNECT message to called gateway **200'** which, in turn, instructs CH **560'** to disconnect its current PSTN connection for this call. To signify that the called gateway has in fact released the

PSTN channel that previously carried this call, CH **560'** issues, as represented by line **1866**, a Q.931 RELEASE message to the PSTN. The PSTN then issues, as represented by line **1869**, a Q.931 RELEASE message to the calling gateway. Once CH **560** releases its PSTN connection for this call, that call handler issues, as represented by line **1872**, a Q.931 RELEASE COMPLETE message to the PSTN which, in turn, issues a corresponding Q.931 RELEASE COMPLETE message to the called gateway and to CH **560'** therein to fully acknowledge release of its PSTN connection for this call.

4. Inter-domain call routing sequence

a. routing information resides in same administrative domain as calling endpoint

FIG. **19** depicts a sequence of inter- and intra-gateway operations **1900** that would occur for routing a telephone call over the data network between two administrative domains in an H.323 environment, e.g., that shown in FIG. **4B**, where the routing information for the called endpoint has been cached within and is supplied by a border element within the same domain as the calling endpoint ("simple call routing"). As noted above, routing information is supplied in the form of a corresponding descriptor as each endpoint registers with a gatekeeper. Once a new endpoint registers with a gatekeeper, that gatekeeper supplies the descriptor of that new endpoint to a border element in same administrative domain as the gatekeeper for publication to all other such gatekeepers in the same domain. In addition, an external border element in one domain can request, from an external border element in another domain, all descriptors for that other domain for internal storage at the former border element. In addition and as noted above, an external border element also caches, for subsequent use, in its local storage descriptors of inter-domain calls that have been recently routed through that border element.

Here, assume that a call is being initiated through a calling directory number associated with a telephony endpoint (not shown) serviced through Administrative Domain A to a called directory number associated with a telephony endpoint (also not shown) serviced through Administrative Domain B. First, calling gateway **200** within this domain, sends, as represented by line **1910**, an H.225.0 admission request (ARQ) message to gatekeeper $\mathbf{420}_1$ that services the calling endpoint. The gatekeeper then determines, as indicated in block **1915**, whether, in the context of routing the call, the called endpoint lies in the same administrative domain, i.e., Domain A, as the calling endpoint. If not, as is the case here, gatekeeper $\mathbf{420}_1$ issues, as represented by line **1920**, an Access Request, containing, e.g., a called directory number for this call, to border element **430** in order to resolve this number into a destination network address for the called endpoint. Since in this scenario, border element **430** possesses the requisite routing information, this border element returns, as represented by line **1930**, an Access Confirm message containing the resolved destination address, i.e., the network address. Gatekeeper **4201**, in response to receiving this address, issues, as represented by line **1940**, an H.225.0 Admission Confirm (ACF) message containing the destination address back to calling gateway **200** which then proceeds, as represented by line **1950**, to interact with called gateway **200'** to establish the call over the data network.

b. routing information resides in different administrative domain as calling endpoint

FIG. **20** depicts a sequence of inter- and intra-gateway operations **2000**, similar to those shown in FIG. **19**, that would occur for routing a telephone call over the data network between two administrative domains, but where the routing information for the called endpoint does not then reside in a border element in the same administrative domain as the calling endpoint.

Here, too, assume that a call is being initiated through a calling directory number associated with a telephony endpoint (not shown) serviced through Administrative Domain A to a called directory number associated with a telephony endpoint (also not shown) serviced through Administrative Domain B. First, calling gateway **200** within this domain, sends, as represented by line **2010**, an H.225.0 admission request (ARQ) message to gatekeeper $\mathbf{420}_1$ that services the calling endpoint. The gatekeeper then determines, as indicated in block **2015**, whether, in the context of routing the call, the called endpoint lies in the same administrative domain, i.e., Domain A, as the calling endpoint. If not, as is the case here, gatekeeper $\mathbf{420}_1$ issues, as represented by line **2020**, an Access Request, containing, e.g., a called directory number for this call, to border element **430** in order to resolve this number into a destination network address for the called endpoint. Since in this scenario, border element **430** does not then possess the requisite routing information, i.e., it does not possess a corresponding descriptor for the called endpoint, this element concludes, as indicated by block **2025**, that it requires a descriptor for this endpoint. Consequently, border element **430** issues, as represented by line **2030**, an Access Request to an external border element, e.g., border element **450**, in the called administrative domain to resolve the called directory number. In response, border **450** element returns, as represented by line **2040**, an Access Confirm message containing the resolved destination address, i.e., the network address to border element **430** which, in turn, updates its own routing table with this descriptor and also, as represented by line **2050**, sends this Access Confirm message to gatekeeper **4201**. Gatekeeper **4201**, in response to receiving this address, issues, as represented by line **2060**, an H.225.0 Admission Request Confirm (ACF) message containing the destination address back to calling gateway **200** which then proceeds, as represented by line **2070**, to interact with called gateway **200'** to establish the call over the data network.

c. Intra-gatekeeper operations

FIG. **26** depicts inter-process interactions **2600** that occur within a gatekeeper, such as gatekeeper **700**, to route a VoIP call being made by a gateway, here gateway **200**, registered with that gatekeeper.

As shown, in response to an incoming H.225.0 admission request (ARQ) message, as represented by line **2610**, from local H.323 process **563** (not shown) executing within gateway **200**, endpoint manager **750**, sends, as represented by line **2610**, an ARQ message to H.323 endpoint **4051**. As noted above, manager **750** manages H.323 endpoints which includes registration and de-registration of endpoints, allocating and de-allocating network bandwidth associated with a call, call routing between endpoints and appropriate endpoint address translation for use by routing process **760**. Hence, once the endpoint manager, within the gatekeeper, receives the ARQ message—which includes the called directory number, that manager determines which particular endpoint, here, e.g., endpoint **4051**, is requesting this call and then forwards, as represented by line **2620**, the ARQ message to it. In response to the ARQ message, endpoint $\mathbf{405}_1$ issues, as represented by line **2630**, a Route Request message to routing process **760** to request destination routing information to the called directory number. In response to this message, routing process **760** examines its associated routing tables to determine whether those tables contain an

entry for the called directory number. If routing information is found, process **760** returns, as represented by line **2640**, a Route OK message containing all possible telephony endpoints that are qualified to accept the call. Once this occur, endpoint 405_1 issues, as represented by line **2650**, a Reserve Bandwidth command to the endpoint manager to reserve an amount of bandwidth as required for this call. Depending on the specific type of communication to be carried over this call, i.e., voice, modem data or facsimile, bandwidth requirements will vary accordingly. Given that available bandwidth is a finite resource, limits are placed on which telephony endpoints can use it and how much of that bandwidth each can use at any specific time. If the requested bandwidth can be allocated for this call, then it is suitably reserved by endpoint manager **750** which, in turn, returns a Bandwidth Reserved message, as represented by line **2660**, to the requesting endpoint. In response, the endpoint then issues, as represented by line **2670**, an H.323 admission confirm (ACF) message to the endpoint manager signifying that the call can be completed over the reserved bandwidth. The endpoint manager, in turn, passes the ACF message to gateway **200** that made the call request and specifically through local H.323 process **563** (not shown) executing within that gateway,

5. Service establishment sequence

FIG. **21** depicts inter-process operational sequence **2100** that occurs between a gatekeeper, e.g., gatekeeper **4051**, and a border element, e.g., border element **430**, in the same administrative domain, for establishing a service relationship therebetween. This typically occurs upon power-up of a gatekeeper or whenever a gatekeeper, previously taken out of service, is then returned to active service.

First, requesting gatekeeper 405_1 issues, as represented by line **2105**, a Service Request message to border element **430**. In response, the border element replies, as represented by line **2110**, if it can provide requested service with a Service Confirm message back to the requesting gatekeeper. Thereafter, as indicated by block **2115**, gatekeeper 405_1 downloads all its own zone-based call routing capabilities to the border element, thereby updating the border element with the routing information for the zone handled by the gatekeeper. To do so, the border element issues, as represented by line **2120**, a Descriptor ID Request message to the gatekeeper in order to obtain an identifier for each descriptor then stored in the gatekeeper. In response, the gatekeeper provides, as represented by line **2125**, a Descriptor ID Confirm message that contains a list of identifiers (Descriptor IDs) for all descriptors then stored within the gatekeeper. Once this information has been received, the border element separately requests and, in response, obtains each descriptor identified in the list. Specifically, for each descriptor requested, border element **430** issues, as represented by line **2130**, a Descriptor Request specifying a corresponding Descriptor ID for that descriptor. In response, the gatekeeper downloads, as represented by line **2135**, the requested descriptor to the border element. The operations represented by lines **2130** and **2135** are iteratively repeated for each successive descriptor requested by the border element.

Once gatekeeper 405_1 has downloaded all its descriptors to border element **430**, the gatekeeper requests, as indicated by block **2140**, administrative zone-based routing information from the border element, i.e., descriptors, associated with all other zones, other than that for gatekeeper **405**, in the same administrative domain as the border element. Hence, once this information has been stored in gatekeeper 405_1, this gatekeeper will possess descriptors for all the endpoints in its domain. Specifically, to do so, the gatekeeper issues, as represented by line **2145**, a zone-based Descriptor ID Request message to the border element in order to obtain an identifier for each descriptor then stored in the border element. In response, the border element provides, as represented by line **2150**, a Descriptor ID Confirm message that contains a list of identifiers for each and every descriptor, other than those provided by gatekeeper 405_1, then stored within that border element. Once this information has been received, the gatekeeper separately requests and, in response, obtains each descriptor identified in the list. Specifically, for each descriptor requested, gatekeeper 405_1 issues, as represented by line **2155**, a Descriptor Request specifying a corresponding Descriptor ID for that descriptor. In response, the border element downloads, as represented by line **2160**, the requested descriptor to the border element. The operations represented by lines **2155** and **2160** are iteratively repeated for each successive descriptor requested by the gatekeeper. Once all the border element provides all its requested routing information to the gatekeeper, a service relationship exists between the two and sequence **2100** ends.

6. Information transfer sequence

FIG. **22** depicts inter-process sequence **2200** that occurs to transfer routing information from one gatekeeper to another in the same administrative domain. As noted above, routing information stored in a gatekeeper changes as gateways and telephony endpoints serviced through that gatekeeper register and de-register themselves with that gatekeeper. As each such element registers and de-registers with its corresponding gatekeeper in an administrative domain, that gatekeeper sends associated routing changes, i.e., descriptor updates, to an associated border element which then distributes the routing changes to all other gatekeepers in that same domain (including gatekeepers that have established service relationships with peer border elements in that domain).

As shown in FIG. **22**, assume a change, as indicated in block **2210**, in a descriptor in a zone serviced by gatekeeper 420_1 occurs, regardless of whether it occurs through registration or de-registration of a gateway or telephony endpoint. Thereafter, as represented by line **2220**, gatekeeper 420_1 issues a Descriptor Update message containing an instruction to update existing descriptor(s) along with appropriate descriptor change(s). These changes include deleting or updating an existing descriptor(s), or adding a new descriptor(s). Once this message is received and processed by border element **430**, that element issues, as represented by line **2230**, a Descriptor Update Confirm message back to gatekeeper 420_1 to acknowledge that the particular descriptor(s) has been updated.

Then, border element **430** issues, as represented by line **2240**, a Descriptor Update message to distribute this descriptor change(s) to a next successive gatekeeper in the domain, here gatekeeper 420_2. Once this message is received and processed by that next gatekeeper, that gatekeeper issues, as represented by line **2250**, a Descriptor Update Confirm message back to border element **430** to acknowledge that the particular descriptor(s) has been updated. Operations **2240** and **2250** are iteratively repeated for every gatekeeper in the administrative domain other than that, e.g., gatekeeper 420_1, which provided the descriptor changes to border element **430**.

Whether the descriptors being updated reside in the border element or in a gatekeeper, the Descriptor Update message communicated to either element contains a sequence of descriptor changes that encompasses all the descriptor changes then needed to be made.

7. Call teardown

a. Inter-gatekeeper operations

FIG. **23** depicts inter-process interactions **2300** that occur between two gatekeepers, here illustratively gatekeepers 420_1 and 420_2, for tearing down a call then carried over the data network (VoIP call).

Specifically, assume that a data call has been established between corresponding telephony endpoints connected to calling and called gateways **200** and **200'**, respectively, and that a user stationed at a calling endpoint serviced by PBX **14** then hangs up his(her) telephone. As a result, this PBX sends, as represented by line **2310**, a Q.931 Disconnect message, containing the calling directory number (here, e.g., "1-732-872-8020") to gateway **200**. This signaling information is conveyed through DTMF (dual-tone multi-frequency) tones, pulses or ISDN D-channel information, as appropriate, over an incoming trunk in T1 trunks **1213**. Gateway **200**, in turn, sends, as represented by line **2320** and over RAS channel **1217**, an H.225.0 disengage request (DRQ) message to a gatekeeper, e.g., gatekeeper 420_1, servicing this endpoint. This DRQ message specifies the dialed number for which this data connection is being torn down by the gateway. In response to the DRQ message, gatekeeper **4201**, returns, as represented by line **2330**, an H.225.0 disengage confirm (DCF) message containing an IP address of the called gateway, since this call is terminated at that gateway. This gatekeeper, as well as peer gatekeeper 460_1, identifies this call through an association previously established between routing information for this call and a unique CallId that was established during setup of this call and reserved for use by this call throughout its entire duration. In response to this DCF message, the calling gateway sends, as represented by line **2340** and via H.245 channel **1223**, a Q.931 Disconnect message, containing the CallId, to the called gateway, here gateway **200'**.

In response to the Disconnect message, called gateway **200'** sends, as represented by line **2350**, an H.225.0 DRQ message to a gatekeeper, here illustrative gatekeeper 460_1, that serves the called endpoint. This message requests this gatekeeper to disconnect the data connection then carrying this call. In response, gatekeeper 460_1 returns, as represented by line **2360** and over RAS channel **1233**, sends an H.225.0 DCF message back to gateway **200'**. In response to this confirmation message, called gateway sends, as represented by line **2370**, a Q.931 Disconnect message over an outgoing trunk in T1 trunks **1247**, to PBX **44** using DTMF, dialing pulses or ISDN, depending on its capability and configuration to disconnect the T1 channel connection to the PBX. Once this PBX releases and frees this channel, PBX **44** then issues, as represented by line **2380**, an appropriate Release Complete message back to the called gateway. This gateway, in turn, issues, as represented by line **2390** and via the H.245 channel, a Release Complete message to the calling gateway. In response to this message, the calling gateway issues, as represented by line **2395**, a Q.931 Release Complete message, via an incoming trunk in T1 trunks **1213**, to PBX **14** to indicate that the connection to the PBX has been completely released for this call. The calling gateway, here gateway **200**, then frees the PBX channel it had been using for this call.

b. Intra-gatekeeper operations

FIG. **27** depicts inter-process interactions **2700** that occur within a gatekeeper, here gatekeeper **700** (not specifically shown in FIG. **27**), to tear down the VoIP call to an endpoint, here, e.g., endpoint 405_1 served by that gatekeeper.

As shown, in response to an incoming H.323 disengage request (DRQ) message, as represented by line **2710**, from local H.323 process **563** (not shown) executing within gateway **200**, and originating with an endpoint registered with gatekeeper **700**, endpoint manager **750**, sends, as represented by line **2720**, the DRQ message to H.323 endpoint 405_1. This message indicates that the endpoint desires to release resources it had been using for an active VoIP call. As noted above, manager **750** manages H.323 endpoints which includes, inter alia, registration and de-registration of endpoints, allocating and de-allocating network bandwidth associated with a call. In response to the DRQ message, the endpoint manager identifies which particular endpoint, here endpoint **4051**, is handling this call and forwards, as represented by line **2720**, the DRQ message to that endpoint. In response, endpoint 405_1 identifies the call instance for this call and issues, as represented by line **2730**, an instruction to free the bandwidth which had been allocated to that endpoint for this call. Once the manager frees this bandwidth for allocation to other calls, the manager sends, as represented by line **2740**, a Bandwidth Freed message to the endpoint signifying that the bandwidth has been freed. In response, the endpoint issues, as represented by line **2750**, an H.323 Disengage Confirm (DCF) message to manager **750** which, in turn and as represented by line **2760**, passes the H.323 DCF message to gateway **200** that issued the request to terminate the call and specifically through local H.323 process **563** (not shown) executing within that gateway.

8. Registration

As discussed above, a gateway must register with a gatekeeper in order for calls to be placed through that gateway. Typically, a gateway registers upon its power-up or being reset, or upon its being returned to service.

Registration serves multiple purposes. First, it establishes service relationships between telephony endpoints and gateways and, for registered endpoints, starts a keep-alive operation during which each endpoint is continuously polled by an Information Request (IRQ) message issued by the gatekeeper for current call routing information. If a registered endpoint in any zone fails to respond, the gatekeeper responsible for that zone de-registers that endpoint and purges its stored call routing information for that endpoint.

Endpoints also register with gatekeepers in accordance with the H.323 standard in order for each such endpoint to supply its routing descriptor to its servicing gatekeeper and thereby originate and terminate calls over the data network. From a standpoint of establishing routing information under H.323, actual telephony endpoints and gateways are registered in the same exact manner and are collectively viewed as "endpoints" with a routing descriptor created and published for each by a corresponding gatekeeper, and the same underlying operations being performed to register either. Consequently, we will treat both as being "endpoints" for purposes of registration and later, in the context of discussing FIGS. **25** and **29**, de-registration.

As discussed above, corresponding routing information, i.e., routing descriptors, for an endpoint (whether it is a terminal or a gateway) provided by a gatekeeper to its external border element is also provided by that border element to its peer border element for use in dynamically updating its routing information such that peered border elements maintain identical routing information.

Since routing information obtained during registration of an H.323 entity is not published by a border element beyond its own administrative domain and hence has no impact on routing information stored in any other domain, we will only discuss those operations that occur within the domain into which that entity, e.g., here, e.g., gateway **200**, is registering.

a. Inter-gatekeeper operations

FIG. **24** depicts inter-process interactions **2400** that occur, in accordance with our invention, in the course of registering a gateway, e.g., gateway **200**, with a gatekeeper, e.g., gatekeeper 420_1.

First, gateway **200** issues, as represented by line **2410** and over RAS channel **1217**, an H.323 Gatekeeper Request (GRQ) message over a gatekeeper multicast address. This message is used to identify all gatekeepers in a zone. In response to this message, a gatekeeper, e.g., gatekeeper 420_1, replies, as represented by line **2420** with an H.323 Gatekeeper Confirm (GCF) message to indicate that it can provide service to the requesting gateway. Thereafter, gateway **200** sends, as represented by line **2430**, an H.323 Registration Request (RRQ) message, including its routing information, to this gatekeeper to specify that this gateway seeks to be serviced by this gatekeeper. If gatekeeper 420_1 is able to service gateway **200** and hence permits that gateway to register, gatekeeper 420_1 issues, as represented by line **2440**, an H.323 Registration Confirm (RCF) message over RAS channel **1217** to the gateway. If registration is denied for any reason, the gatekeeper sends an H.323 Registration Reject message (RRJ—not shown) back to the gatekeeper which, in turn, terminates the registration process.

Should gatekeeper 420_1 confirm the registration request, that gatekeeper issues, as represented by line **2450** and over service channel **2455**, a Descriptor Update message to its associated border element, here element **430**. This message contains the routing information just supplied by the registering gateway. In response, border element **430** sends, as represented by line **2460**, a Descriptor Update Acknowledgement message over this service channel and back to gatekeeper 420_1, to confirm receipt of this routing information. Furthermore, border element **430** issues, as represented by line **2470** and over control channel **2465**, an Information Update message to its peer border element, here element **430'**. This update message contains the routing information supplied by the registering gateway and is used by the peer border element to update its routing information. Hence, both peered border elements maintain the same routing information across their entire and common administrative domain.

Thereafter, border element **430** publishes the updated routing information, i.e., routing descriptor, it just obtained from gatekeeper 420_1 to every other gatekeeper in its administrative domain. To do so, border element **430** issues, as represented by line **2480** and over service channel **2485**, a Descriptor Update message to a next successive gatekeeper, here gatekeeper 420_2, in its administrative domain. This update contains the routing descriptor provided by and for newly registered gateway **200**. Once gatekeeper 420_2 receives this update, it responds with a Descriptor Update Acknowledge message, as represented by line **2490** over service channel **2485**, back to border element **430**. Gatekeeper 420_2 then updates its own routing information accordingly. If more than one other gatekeeper exists in the administrative domain, then border element **430** issues a separate Descriptor Update message to that element and receives therefrom a separate Descriptor Update Acknowledgement message from that element, prior to proceeding to providing the new routing descriptor to the next element, and so forth.

b. Intra-gatekeeper operations

FIG. **28** depicts inter-process interactions **2800** that occur within a gatekeeper for registering a new gateway with that gatekeeper. As noted above, H.323, from a standpoint of registering an endpoint, in the sense of a telephony endpoint and a gateway, collectively views both as being an "endpoints" and hence makes no distinction in registering one over the other in terms of the underlying operations that register these entities.

As shown, endpoint **2805**, here, e.g., registering gateway **200**, issues, as represented by line **2810**, the H.323 Registration Request (RRQ) message to the gatekeeper. In response, endpoint manager **750**, through querying its routing information, determines whether routing information is presently stored for new endpoint **2805**. If not, the gatekeeper creates, as represented by line **2820**, a new endpoint entry, for gateway **200**, in its routing tables for this new endpoint and readies that entry for use. Once this occurs, endpoint manager **750** forwards, as represented by line **2830**, the RRQ message to this new endpoint (i.e., to gateway **200**). In response to the RRQ message, this new endpoint extracts its routing information, i.e., a routing descriptor, from this message, and issues, as indicated by line **2840**, an Add Zone address message, containing this routing descriptor, to Routing process **760**. In response, this process updates its routing tables to insert routing information contained in this descriptor into its zone routing tables. Thereafter, endpoint **2805** issues, as represented by line **2850**, an Endpoint Register command, including its routing descriptor, to administrative domain manager **740** which, in turn, publishes the routing descriptor to all other gatekeepers in the administrative domain. Finally, endpoint manager **750** confirms the registration by issuing, as represented by line **2860**, an H.323 Registration Confirm (RCF) message back to the gateway.

9. De-registration

As noted above, routing information obtained during registration of an H.323 entity is not published by a border element beyond its own administrative domain and hence, from the standpoint of its publication, has no impact on any other domain. The converse is also true during de-registration, in that changes in routing information occasioned during de-registration are not published beyond an administrative domain in which that entity is de-registering. Hence, we confine our discussion of the operations that occur during de-registration to the administrative domain in which an entity, here, e.g., gateway **200**, is de-registering.

a. Inter-gatekeeper operations

FIG. **25** depicts inter-process interactions **2500** that occur, in accordance with our invention, in the course of de-registering a gateway, e.g., gateway **200**, from a gatekeeper.

First, gateway **200** issues, as represented by line **2510** and over RAS channel **1217**, an H.323 Gatekeeper Unregistration Request (URQ) message. This message indicates that this gateway no longer requires service from its servicing gatekeeper. In response to this message, a gatekeeper, e.g., gatekeeper 420_1, replies, as represented by line **2520** with an H.323 Unregistration Confirm (UCF) message indicating that the gateway will no longer be provided with such service.

Gatekeeper 420_1 then issues, as represented by line **2530** and over service channel **2455**, a Descriptor Update message to its associated border element, here element **430**. This message contains the routing information that is to be removed for gateway **200** from the routing tables by both the border elements and the other gatekeepers in the administrative domain. In response, border element **430** sends, as represented by line **2540**, a Descriptor Update Acknowledgement message over this service channel and back to gatekeeper 420_1, to confirm receipt of this routing informa-

tion. Furthermore, border element **430** issues, as represented by line **2550** and over control channel **2465**, an Information Update message to its peer border element, here element **430'**. This update message contains the routing information supplied by the registering gateway and is to be deleted by the peer border from its routing information as well.

Thereafter, border element **430** publishes the deleted routing information, i.e., routing descriptor, it just obtained from gatekeeper 420_1 to every other gatekeeper in its administrative domain. To do so, border element **430** issues, as represented by line **2560** and over service channel **2485**, a Descriptor Update message to a next successive gatekeeper, here gatekeeper 420_2, in its administrative domain. This update contains the routing descriptor associated with gateway **200** that has been de-registered. Once gatekeeper 420_2 receives this update, it responds with a Descriptor Update Acknowledge message, as represented by line **2570** over service channel **2485**, back to border element **430**. Gatekeeper 420_2 then updates its own routing information accordingly. If more than one other gatekeeper exists in the administrative domain, then border element **430** issues a separate Descriptor Update message to that element and receives therefrom a separate Descriptor Update Acknowledgement message from that element, prior to proceeding to the next element that routing descriptor that is now to be deleted, and so forth.

b. Intra-gatekeeper operations

FIG. **29** depicts inter-process interactions **2900** that occur within a gatekeeper for de-registering a gateway, e.g., gateway **200**, from that gatekeeper. As noted above, H.323, from a standpoint of de-registering a telephony endpoint, in the sense of a telephony endpoint and a gateway, collectively views both as being an "endpoints" and hence makes no distinction in registering one over the other in terms of the underlying operations that de-register these entities.

As shown, endpoint **2805**, here, e.g., registering gateway **200**, issues, as represented by line **2910**, the H.323 Unregistration Request (URQ) message to the gatekeeper. In response, endpoint manager **750**, through querying its routing information, locates the endpoint in its routing tables and forwards, as represented by line **2920**, the URQ message to this existing endpoint (i.e., to gateway **200**). In response to the URQ message, the endpoint issues, as indicated by line **2930**, a Delete Zone Address containing its routing descriptor, to Routing process **760**. Thereafter, this process updates its routing tables by deleting the routing information contained in this descriptor from its zone routing tables. Thereafter, endpoint **2805** issues, as represented by line **2940**, an Endpoint Unregister command, including its routing descriptor, to administrative domain manager **740** which, in turn, publishes the routing descriptor to all other gatekeepers in the administrative domain such that routing information in this descriptor can be deleted from all other routing tables maintained throughout the domain. Endpoint manager **750** then confirms the de-registration by issuing, as represented by line **2950**, an H.323 Unregistration Confirm (UCF) message back to the gateway. Finally, endpoint manager **750** removes, as represented by line **2960**, the endpoint.

Though we have described our inventive gateway as functioning with digital PBXs that rely on T1/E1 connections, the interfaces to the gateway can be readily modified by anyone skilled in the art to accommodate a wide range of different telephone subscriber line types and speeds, including analog plain old telephone (POTs) connections, DSL (digital subscriber line)—including ADSL (asymmetric DSL), and ISDN (integrated service digital network) connections.

Moreover, though we have described border elements as dynamically constructing their internal call routing tables based on on-going registration and deregistration of gateways and telephony endpoints in their corresponding administrative domains, these routing tables could alternatively be statically configured for each border element to simplify processing by eliminating or reducing a need for each border element to update its routing tables as a gateway(s) and/or other border element(s) establishes a service relationship with it and/or in response to an endpoint registering or de-registering with a gatekeeper in its domain, thus potentially expediting call routing through that former border element by eliminating latency associated with such updates.

Although a single, though rather detailed, embodiment which incorporates the teachings of the present invention has been shown and described in considerable detail herein, those skilled in the art can readily devise many other embodiments that still utilize these teachings.

We claim:

1. Apparatus for a telephony gateway for routing a telephony call through either a public-switched telephone network (PSTN) or a data network to a peer telephony gateway, comprising:

(A) a processor;

(B) a memory, connected to the processor and storing computer executable instructions therein; and

(C) circuitry, controlled by and connected to the processor, for interfacing the gateway to the PSTN and the data network;

(D) wherein the processor, in response to performing the executable instructions:

(D1) establishes, through the interface, a connection, through one of the PSTN and the data network, with the peer gateway and over which the call is to be carried;

(D2) exchanges call specific data with the peer gateway, the data containing both a call identifier common to both the gateway and the peer gateway, the call identifier uniquely distinguishing the call from all other calls then being handling by the gateway or the peer gateway wherein both the gateway and the peer gateway form a same association between the call and the call identifier;

(D3) dynamically measures at least one predefined characteristic of a data network connection extending from the gateway to the peer gateway, throughout a duration of the call so as to define quality of service (QoS); and

(D4) in response to a sufficient increase or decrease in the QoS, issues an appropriate message, containing the call identifier, to the peer gateway and, as a result thereof, interacts with the peer gateway to establish a connection through the other one of the PSTN and the data network and then switches the call between the PSTN and the data network, such that the call is alternately and automatically switched back and forth between the PSTN and the data network through interactions of the gateway and the peer gateway responsive to dynamic changes in the QoS.

2. The apparatus in claim **1** wherein the connection is first established over the data network.

3. The apparatus in claim **1** wherein the predefined characteristic comprises packet loss, latency or jitter.

4. The apparatus in claim **1** wherein the processor, in response to execution of the stored instructions, exchanges the call specific data while the connection is being established.

5. The apparatus in claim 4 wherein the call specific information is exchanged over the connection through the PSTN or the data network.

6. The apparatus in claim 1 wherein the call specific data comprises an indication that the call can be automatically switched by both the gateway and the peer gateway between the PSTN and the data network; and wherein the processor, in response to the execution of the instructions, issues the appropriate message if the indication specifies that the call can be switched by both the gateway and the peer gateway.

7. The apparatus in claim 6 wherein the connection is first established over the data network.

8. The apparatus in claim 6 wherein the processor, in response to execution of the stored instructions:

(a) should the QoS decrease below a predefined threshold, establishes a PSTN connection through the PSTN and specifies to the peer gateway and over the PSTN connection, the call identifier, such that, in response, the gateway and the peer gateway both switch the call from the data network connection to the PSTN connection and then terminate the data network connection; and

(b) while the gateway routes the call over the PSTN connection, should the QoS subsequently equal or exceed the threshold, re-establishes the connection through the data network and specifies to the peer gateway and over the re-established data network connection, the call identifier, such that, in response, the gateway and the peer gateway both switch the call from the PSTN to the re-established data network connection and then terminate the PSTN connection; and

(c) repeats operations (a) and (b) throughout the call to switch the call back and forth between the data network and the PSTN in response to the dynamic changes in the QoS.

9. The apparatus in claim 8 wherein the predefined characteristic comprises packet loss, latency or jitter.

10. The apparatus in claim 8 wherein the processor, in response to execution of the stored instructions:

to initiate the call through the gateway, generates a set up message to set up the call and transmits the set up message to the peer gateway; and

receives a responding message, from the peer gateway and issued in response to receipt of the set up message, the responding message containing the call identifier with the call identifier having been generated by the peer gateway.

11. The apparatus in claim 10 wherein the processor, in response to execution of the stored instructions and receipt of the call identifier in the responding message, forms the association between the call and the identifier for subsequent use in switching the call between the PSTN and the data network.

12. The apparatus in claim 11 wherein the processor, in response to execution of the stored instructions and prior to issuing the setup message to switch the call to the data network, determines that sufficient bandwidth is available to support the call and an associated calling endpoint has sufficient permission to access the data network for the call.

13. The apparatus in claim 12 wherein the setup message comprises an H.323 SETUP message and the responding message comprises either an H.323 CALL PROCEEDING or H.323 CONNECT message.

14. The apparatus in claim 10 wherein the setup message contains a calling flag, the calling flag comprising the indication as to whether the gateway can switch the call between the PSTN and data network.

15. The apparatus in claim 14 wherein the calling flag, is contained within a "nonstandard data" field in the H.323 SETUP message.

16. The apparatus in claim 14 wherein the responding message contains a called flag, the call identifier and a pooled directory number associated with the peer gateway.

17. The apparatus in claim 16 wherein the called flag, the call identifier and the pooled directory number are all contained within a "nonstandard data" field in either the H.323 CALL PROCEEDING or H.323 CONNECT messages.

18. The apparatus in claim 16 wherein the call specific data is contained within a predetermined H.323 message with the data being carried using a call independent signaling feature of the predetermined message.

19. The apparatus in claim 16 wherein the processor, in response to execution of the stored instructions, establishes the PSTN connection to the peer gateway over the pool directory number specified in the responding message.

20. The apparatus in claim 8 wherein, for predefined called telephone numbers, the indication is set to an appropriate value specifying that any call made to any of said predefined called telephone numbers through the gateway can not be switched by the gateway between the PSTN and the data network.

21. The apparatus in claim 6 wherein the processor, in response to execution of the stored instructions, receives an appropriate message, containing the call identifier and originating from the peer gateway, to switch the call between the PSTN and the data network and which results from a sufficient increase or decrease in the QoS as dynamically measured by the peer gateway.

22. The apparatus in claim 21 wherein the processor, in response to the appropriate message received from the peer gateway switches the call between the PSTN and the data network.

23. The apparatus in claim 1 wherein the processor, in response to execution of the stored instructions:

(a) should the QoS decrease below a predefined threshold, establishes a PSTN connection through the PSTN and specifies to the peer gateway and over the PSTN connection, the call identifier, such that the gateway and the peer gateway both switch the call from the data network connection to the PSTN connection, and then terminate the data network connection; and

(b) while the gateway routes the call over the PSTN connection, should the QoS subsequently equal or exceed the threshold, re-establishes the connection through the data network and specifies to the peer gateway and over the re-established data network connection, the call identifier, such that the gateway and the peer gateway both switch the call from the PSTN to the re-established data network connection, and then terminate the PSTN connection; and

(c) repeats operations (a) and (b) throughout the call to switch the call back and forth between the data network and the PSTN in response to the dynamic changes in the QoS.

24. The apparatus in claim 23 wherein the predefined characteristic comprises packet loss, latency or jitter.

25. The apparatus in claim 23 wherein the processor, in response to execution of the stored instructions:

to initiate the call through the gateway, generates a set up message to set up the call and transmits the set up message to the peer gateway; and

receives a responding message, from the peer gateway and issued in response to receipt of the set up message, the responding message containing the call identifier with the call identifier having been generated by the peer gateway.

26. The apparatus in claim 25 wherein the processor, in response to execution of the stored instructions and receipt of the call identifier in the responding message, forms the association between the call and the identifier for subsequent use in switching the call between the PSTN and the data network.

27. The apparatus in claim 26 wherein the processor, in response to execution of the stored instructions and prior to issuing the setup message to switch the call to the data network, determines that sufficient bandwidth is available to support the call and an associated calling endpoint has sufficient permission to access the data network for the call.

28. The apparatus in claim 27 wherein the setup message comprises an H.323 SETUP message and the responding message comprises either an H.323 CALL PROCEEDING or H.323 CONNECT message.

29. The apparatus in claim 25 wherein the setup message contains a calling flag, the calling flag comprising the indication as to whether the gateway can switch the call between the PSTN and data network.

30. The apparatus in claim 29 wherein the calling flag, is contained within a "nonstandard data" field in the H.323 SETUP message.

31. The apparatus in claim 29 wherein the responding message contains a called flag, the call identifier and a pooled directory number associated with the peer gateway.

32. The apparatus in claim 31 wherein the called flag, the call identifier and the pooled directory number are all contained within a "nonstandard data" field in either the H.323 CALL PROCEEDING or H.323 CONNECT messages.

33. The apparatus in claim 31 wherein the call specific data is contained within a predetermined H.323 message with the data being carried using a call independent signaling feature of the predetermined message.

34. The apparatus in claim 31 wherein the processor, in response to execution of the stored instructions, establishes the PSTN connection to the peer gateway over the pool directory number specified in the responding message.

35. The apparatus in claim 1 wherein the processor, in response to execution of the stored instructions:

to initiate the call through the gateway, generates a set up message to set up the call and transmits the set up message to the peer gateway; and

receives a responding message, from the peer gateway and issued in response to receipt of the set up message, the responding message containing the call identifier with the call identifier having been generated by the peer gateway.

36. The apparatus in claim 35 wherein the processor, in response to execution of the stored instructions and receipt of the call identifier in the responding message, forms the association between the call and the identifier for subsequent use in switching the call between the PSTN and the data network.

37. The apparatus in claim 36 wherein the processor, in response to execution of the stored instructions and prior to issuing the setup message to switch the call to the data network, determines that sufficient bandwidth is available to support the call and an associated calling endpoint has sufficient permission to access the data network for the call.

38. The apparatus in claim 37 wherein the setup message comprises an H.323 SETUP message and the responding message comprises either an H.323 CALL PROCEEDING or H.323 CONNECT message.

39. The apparatus in claim 35 wherein the setup message contains a calling flag, the calling flag comprising the indication as to whether the gateway can switch the call between the PSTN and data network.

40. The apparatus in claim 39 wherein the calling flag, is contained within a "nonstandard data" field in the H.323 SETUP message.

41. The apparatus in claim 39 wherein the responding message contains a called flag, the call identifier and a pooled directory number associated with the peer gateway.

42. The apparatus in claim 41 wherein the called flag, the call identifier and the pooled directory number are all contained within a "nonstandard data" field in either the H.323 CALL PROCEEDING or H.323 CONNECT messages.

43. The apparatus in claim 41 wherein the call specific data is contained within a predetermined H.323 message with the data being carried using a call independent signaling feature of the predetermined message.

44. The apparatus in claim 41 wherein the processor, in response to execution of the stored instructions, establishes the PSTN connection to the peer gateway over the pool directory number specified in the responding message.

45. The apparatus in claim 35 wherein the predefined characteristic comprises packet loss, latency or jitter.

46. A method for use in a telephony gateway which routes a telephony call through either a public-switched telephone network (PSTN) or a data network to a peer telephony gateway, comprising the steps of:

establishing, through the interface, a connection, through one of the PSTN and the data network, with the peer gateway and over which the call is to be carried;

exchanging call specific data with the peer gateway, the data containing both a call identifier common to both the gateway and the peer gateway, the call identifier uniquely distinguishing the call from all other calls then being handling by the gateway or the peer gateway wherein both the gateway and the peer gateway form a same association between the call and the call identifier;

dynamically measuring at least one predefined characteristic of a data network connection extending from the gateway to the peer gateway, throughout a duration of the call so as to define quality of service (QoS); and

in response to a sufficient increase or decrease in the QoS, issuing an appropriate message, containing the call identifier, to the peer gateway and, as a result thereof, interacting with the peer gateway to establish a connection through the other one of the PSTN and the data network and then switching the call between the PSTN and the data network, such that the call is alternately and automatically switched back and forth between the PSTN and the data network through interactions of the gateway and the peer gateway responsive to dynamic changes in the QoS.

47. The method in claim 46 wherein the establishing step comprises the step of first establishing the connection over the data network.

48. The method in claim 46 wherein the predefined characteristic comprises packet loss, latency or jitter.

49. The method in claim 46 further comprising the step of exchanging the call specific data while the connection is being established.

50. The method in claim 49 further comprising the step of exchanging the call specific information over the connection through the PSTN or the data network.

51. The method in claim 46 wherein the call specific data comprises an indication that the call can be automatically switched by both the gateway and the peer gateway between the PSTN and the data network; and wherein the processor, in response to the execution of the instructions, issues the appropriate message if the indication specifies that the call can be switched by both the gateway and the peer gateway.

52. The method in claim 51 wherein the establishing step further comprises the step of first establishing the connection over the data network.

53. The method in claim 51 further comprising the steps of:

(a) should the QoS decrease below a predefined threshold, establishing a PSTN connection through the PSTN and specifying to the peer gateway and over the PSTN connection, the call identifier, such that, in response, the gateway and the peer gateway both switch the call from the data network connection to the PSTN connection and then terminate the data network connection; and

(b) while the gateway routes the call over the PSTN connection, should the QoS subsequently equal or exceed the threshold, re-establishing the connection through the data network and specifying to the peer gateway and over the re-established data network connection, the call identifier, such that, in response, the gateway and the peer gateway both switch the call from the PSTN to the re-established data network connection and then terminate the PSTN connection; and

(c) repeating steps (a) and (b) throughout the call to switch the call back and forth between the data network and the PSTN in response to the dynamic changes in the QoS.

54. The method in claim 53 wherein the predefined characteristic comprises packet loss, latency or jitter.

55. The method in claim 53 further comprising the steps of:

to initiate the call through the gateway, generating a set up message to set up the call and transmitting the set up message to the peer gateway; and

receiving a responding message, from the peer gateway and issued in response to receipt of the set up message, the responding message containing the call identifier with the call identifier having been generated by the peer gateway.

56. The method in claim 55 further comprising the step, responsive to receipt of the call identifier in the responding message, of forming the association between the call and the identifier for subsequent use in switching the call between the PSTN and the data network.

57. The method in claim 56 further comprising the step, prior to issuing the setup message to switch the call to the data network, of determining that sufficient bandwidth is available to support the call and an associated calling endpoint has sufficient permission to access the data network for the call.

58. The method in claim 57 wherein the setup message comprises an H.323 SETUP message and the responding message comprises either an H.323 CALL PROCEEDING or H.323 CONNECT message.

59. The method in claim 55 wherein the setup message contains a calling flag, the calling flag comprising the indication as to whether the gateway can switch the call between the PSTN and data network.

60. The method in claim 59 wherein the calling flag, is contained within a "nonstandard data" field in the H.323 SETUP message.

61. The method in claim 59 wherein the responding message contains a called flag, the call identifier and a pooled directory number associated with the peer gateway.

62. The method in claim 61 wherein the called flag, the call identifier and the pooled directory number are all contained within a "nonstandard data" field in either the H.323 CALL PROCEEDING or H.323 CONNECT messages.

63. The method in claim 61 wherein the call specific data is contained within a predetermined H.323 message with the data being carried using a call independent signaling feature of the predetermined message.

64. The method in claim 61 further comprising the step of establishing the PSTN connection to the peer gateway over the pool directory number specified in the responding message.

65. The method in claim 53 further comprising the step setting, for predefined called telephone numbers, the indication to an appropriate value specifying that any call made to any of said predefined called telephone numbers through the gateway can not be switched by the gateway between the PSTN and the data network.

66. The method in claim 51 further comprising the step of receiving an appropriate message, containing the call identifier and originating from the peer gateway, to switch the call between the PSTN and the data network and which results from a sufficient increase or decrease in the QoS as dynamically measured by the peer gateway.

67. The method in claim 66 further comprising the step, in response to the appropriate message received from the peer gateway, of switching the call between the PSTN and the data network.

68. The method in claim 46 further comprising the steps of:

(a) should the QoS decrease below a predefined threshold, establishing a PSTN connection through the PSTN and specifying to the peer gateway and over the PSTN connection, the call identifier, such that the gateway and the peer gateway both switch the call from the data network connection to the PSTN connection, and then terminate the data network connection; and

(b) while the gateway routes the call over the PSTN connection, should the QoS subsequently equal or exceed the threshold, re-establishing the connection through the data network and specifying to the peer gateway and over the re-established data network connection, the call identifier, such that the gateway and the peer gateway both switch the call from the PSTN to the re-established data network connection, and then terminate the PSTN connection; and

(c) repeating steps (a) and (b) throughout the call to switch the call back and forth between the data network and the PSTN in response to the dynamic changes in the QoS.

69. The method in claim 68 wherein the predefined characteristic comprises packet loss, latency or jitter.

70. The method in claim 68 further comprising the steps of:

to initiate the call through the gateway, generating a set up message to set up the call and transmitting the set up message to the peer gateway; and

receiving a responding message, from the peer gateway and issued in response to receipt of the set up message, the responding message containing the call identifier with the call identifier having been generated by the peer gateway.

71. The method in claim 70 further comprising the step, responsive to receipt of the call identifier in the responding message, of forming the association between the call and the identifier for subsequent use in switching the call between the PSTN and the data network.

72. The method in claim 71 further comprising the step, prior to issuing the setup message to switch the call to the data network, of determining that sufficient bandwidth is available to support the call and an associated calling endpoint has sufficient permission to access the data network for the call.

73. The method in claim 72 wherein the setup message comprises an H.323 SETUP message and the responding message comprises either an H.323 CALL PROCEEDING or H.323 CONNECT message.

74. The method in claim 70 wherein the setup message contains a calling flag, the calling flag comprising the indication as to whether the gateway can switch the call between the PSTN and data network.

75. The method in claim 74 wherein the calling flag, is contained within a "nonstandard data" field in the H.323 SETUP message.

76. The method in claim 74 wherein the responding message contains a called flag, the call identifier and a pooled directory number associated with the peer gateway.

77. The method in claim 76 wherein the called flag, the call identifier and the pooled directory number are all contained within a "nonstandard data" field in either the H.323 CALL PROCEEDING or H.323 CONNECT messages.

78. The method in claim 76 wherein the call specific data is contained within a predetermined H.323 message with the data being carried using a call independent signaling feature of the predetermined message.

79. The method in claim 76 further comprising the step of establishing the PSTN connection to the peer gateway over the pool directory number specified in the responding message.

80. The method in claim 46 further comprising the steps of:

to initiate the call through the gateway, generating a set up message to set up the call and transmitting the set up message to the peer gateway; and

receiving a responding message, from the peer gateway and issued in response to receipt of the set up message, the responding message containing the call identifier with the call identifier having been generated by the peer gateway.

81. The method in claim 80 further comprising the step, responsive to receipt of the call identifier in the responding message, of forming the association between the call and the identifier for subsequent use in switching the call between the PSTN and the data network.

82. The method in claim 81 further comprising the step, prior to issuing the setup message to switch the call to the data network, of determining that sufficient bandwidth is available to support the call and an associated calling endpoint has sufficient permission to access the data network for the call.

83. The method in claim 82 wherein the setup message comprises an H.323 SETUP message and the responding message comprises either an H.323 CALL PROCEEDING or H.323 CONNECT message.

84. The method in claim 80 wherein the setup message contains a calling flag, the calling flag comprising the indication as to whether the gateway can switch the call between the PSTN and data network.

85. The method in claim 84 wherein the calling flag, is contained within a "nonstandard data" field in the H.323 SETUP message.

86. The method in claim 84 wherein the responding message contains a called flag, the call identifier and a pooled directory number associated with the peer gateway.

87. The method in claim 86 wherein the called flag, the call identifier and the pooled directory number are all contained within a "nonstandard data" field in either the H.323 CALL PROCEEDING or H.323 CONNECT messages.

88. The method in claim 86 wherein the call specific data is contained within a predetermined H.323 message with the data being carried using a call independent signaling feature of the predetermined message.

89. The method in claim 86 further comprising the step of establishing the PSTN connection to the peer gateway over the pool directory number specified in the responding message.

90. The method in claim 80 wherein the predefined characteristic comprises packet loss, latency or jitter.

* * * * *

US006426955B1

(12) **United States Patent**
Gossett Dalton, Jr. et al.

(10) **Patent No.: US 6,426,955 B1**
(45) **Date of Patent: Jul. 30, 2002**

(54) **INTERNET TELEPHONY CALL ROUTING ENGINE**

(75) Inventors: **James Pleasant Gossett Dalton, Jr.**; **Alcina de Figueiredo Dalton**, both of Atlanta; **Stephen Anthony Thomas**, Marietta; **Maria Osterholt Cown**, Austell; **Rich Carroll Vaughn**, Roswell; **Pawel Szczerbina**, Atlanta; **Brian Keith Lovette**, Duluth, all of GA (US)

(73) Assignee: **Transnexus, Inc.**, Atlanta, GA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/154,564**

(22) Filed: **Sep. 16, 1998**

Related U.S. Application Data

(60) Provisional application No. 60/059,087, filed on Sep. 16, 1997.

(51) **Int. Cl.**[7] **H04L 12/28**

(52) **U.S. Cl.** **370/401**; 370/352; 379/133; 379/114

(58) **Field of Search** 370/401, 400, 370/351, 352, 353, 356, 230, 252, 259, 261, 267, 271; 709/223, 224, 227, 228; 379/111, 112, 133, 220, 221, 86.13, 258, 243, 244

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,434,848 A 7/1995 Chimento, Jr. et al. 370/17

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

WO	WO 97/14236	4/1997
WO	WO 97/23078	6/1997
WO	WO 98/36543	8/1998

OTHER PUBLICATIONS

Thom, "H.323: The Multimedia Communications Standard for Local Area Networks," IEEE Communications Magazine, Dec. 1996, pp. 52–56.

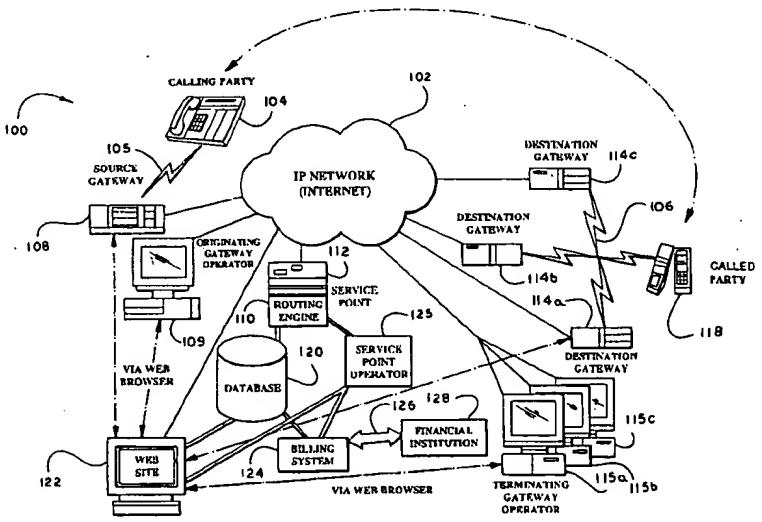
(List continued on next page.)

Primary Examiner—Douglas Olms
Assistant Examiner—Ricardo M. Pizarro
(74) *Attorney, Agent, or Firm*—King & Spalding

(57) **ABSTRACT**

The present invention discloses a centralized routing engine that is able to assist gateways in making routing decisions for calls being placed in an IP network environment. Types of calls include voice, fax, video, etc. The routing engine provides significant flexibility to the gateways by allowing the gateways to designate preferences that define operational limits or requirements. A source gateway operator may set preferences such as the maximum price that is willing to be paid for a call, the maximum delay that will be tolerated and the maximum autonomous system hop count that will be tolerated. A destination gateway operator is likely only to be concerned with setting price schedules as preferences. Gateway operators may also set 'preference criteria,' which define the circumstances in which a certain set of preferences is to be applied. Based on preferences and preference criteria, the routing engine is able to locate destination gateways that are eligible to terminate a voice over telephony IP call. The routing engine provides a prioritized list of eligible destination gateways to the source gateway. The source gateway then works through the prioritized list and attempts to set up the voice over IP telephony call with each eligible destination gateway, until the call is established.

49 Claims, 12 Drawing Sheets



U.S. PATENT DOCUMENTS

5,943,657	A	*	8/1999	Freestone	705/400
5,966,427	A	*	10/1999	Shaffer	379/15.05
6,005,925	A	*	12/1999	Johnson	379/112
6,005,926	A	*	12/1999	Mashinsky	379/114
6,049,531	A	*	4/2000	Roy	370/260
6,128,304	A	*	10/2000	Gardell	370/401
6,178,510	B1	*	1/2001	O'Connor	713/201
6,240,449	B1	*	5/2001	Nadeau	709/223

OTHER PUBLICATIONS

Rudkin, et al., "Real-time applications on the Internet," BT Technology Journal, vol. 15, No. 2, Apr. 1997, pp. 209–225.

The Ascend Max Voice Gateway, XP-002096239, "The asnet pipeline," www.asnet.co.nz/pipeline/sum97/tam-vg.html, Mar. 11, 1999.

Hansson, et al., "Phone Doubler—A step towards integrated Internet and telephone communities," Ericsson Review No. 4, 1997, pp. 142–151.

* cited by examiner

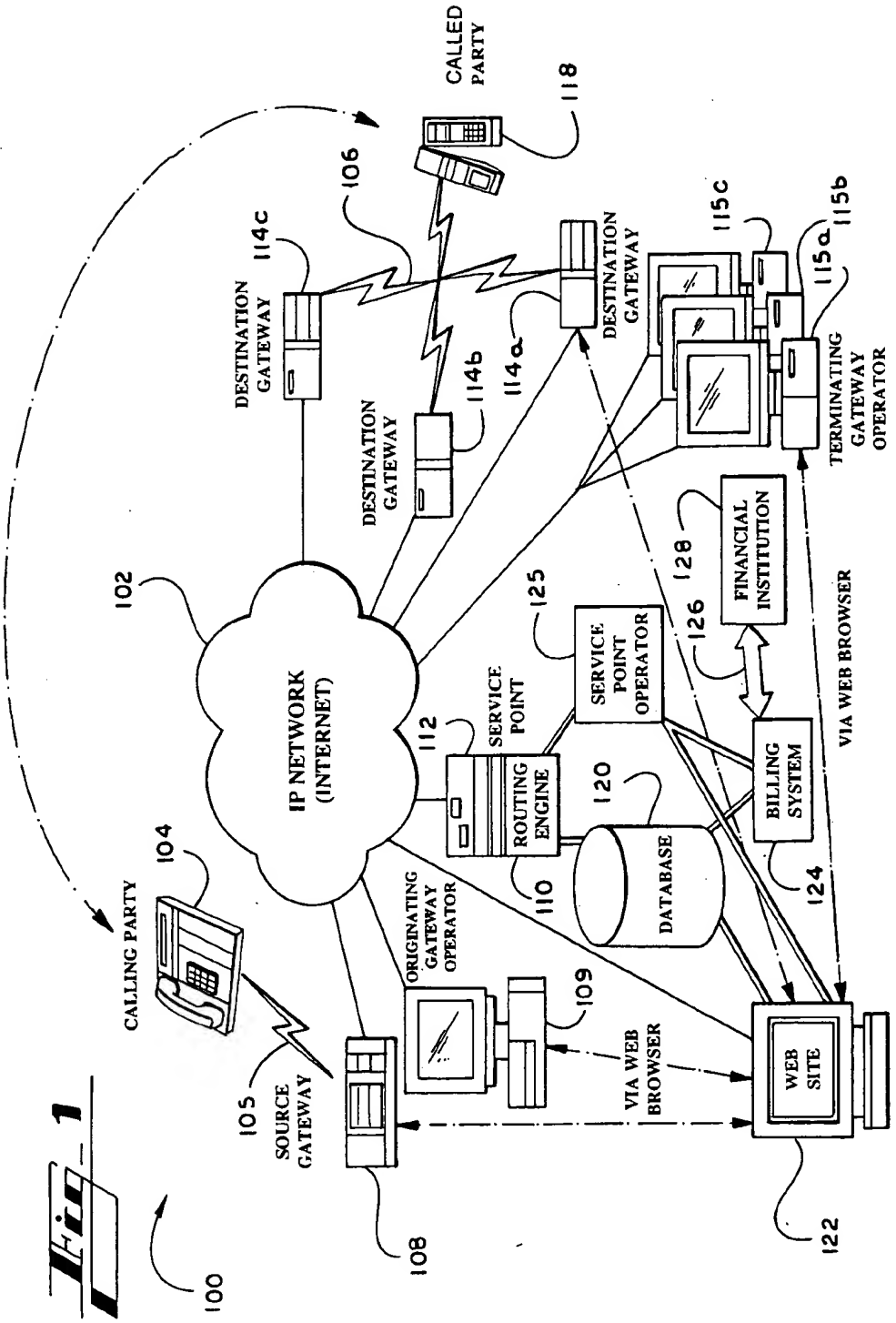

CALLED PARTY
118
106
DESTINATION GATEWAY
114c
DESTINATION GATEWAY
114b
DESTINATION GATEWAY
114a
115c
115a
115b
TERMINATING GATEWAY OPERATOR
FINANCIAL INSTITUTION
128
125
SERVICE POINT OPERATOR
126
VIA WEB BROWSER
102
IP NETWORK (INTERNET)
SERVICE POINT
112
ROUTING ENGINE
110
120
DATABASE
BILLING SYSTEM
124
104
CALLING PARTY
ORIGINATING GATEWAY OPERATOR
109
VIA WEB BROWSER
WEB SITE
105
SOURCE GATEWAY
108
122
100

Fig. 1

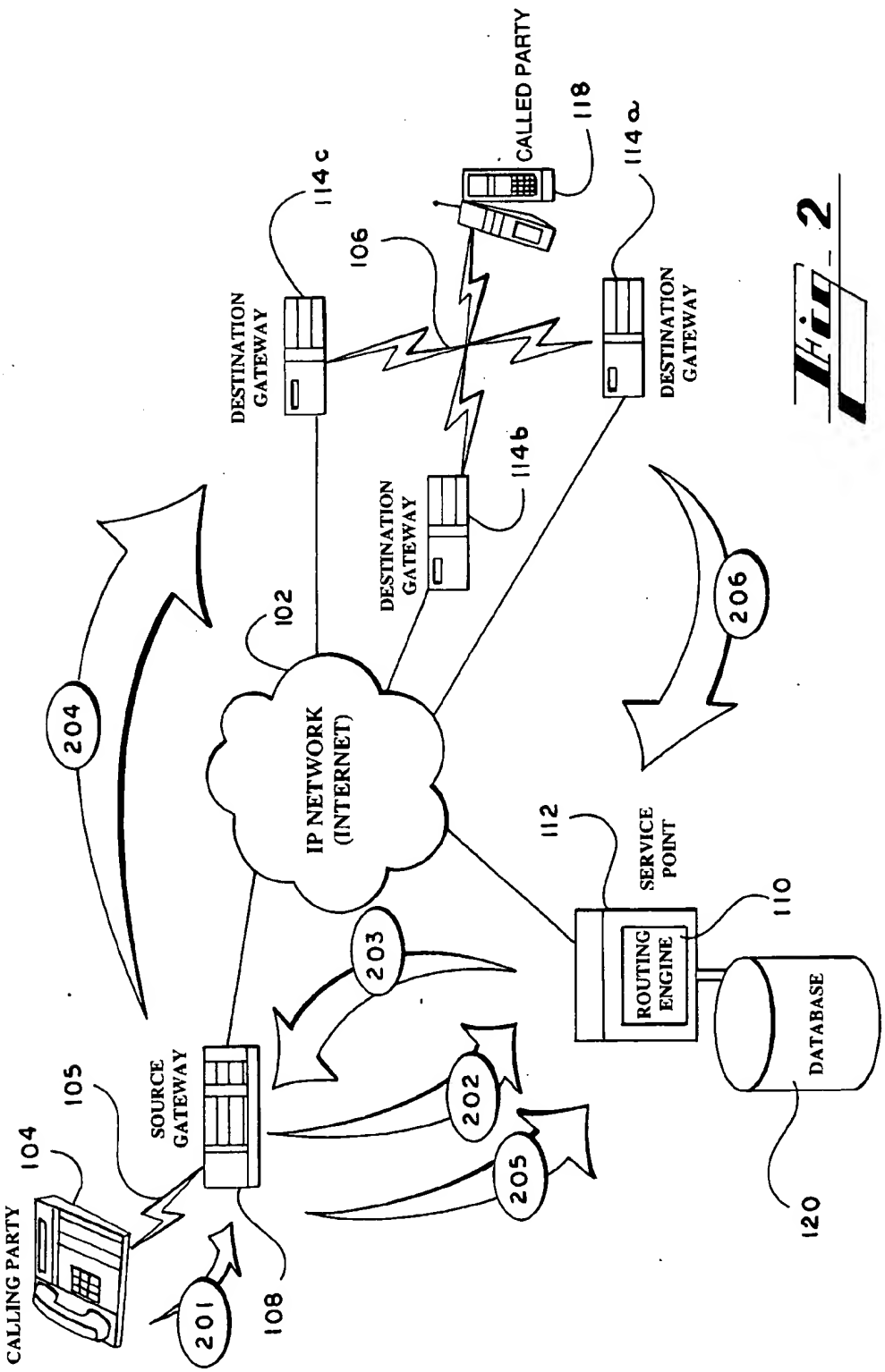

CALLED PARTY
118
114c
114a
106
DESTINATION GATEWAY
DESTINATION GATEWAY
DESTINATION GATEWAY
114b
102
IP NETWORK (INTERNET)
204
206
112
SERVICE POINT
110
ROUTING ENGINE
DATABASE
203
202
205
105
SOURCE GATEWAY
104
CALLING PARTY
201
108
120

Fig. 2

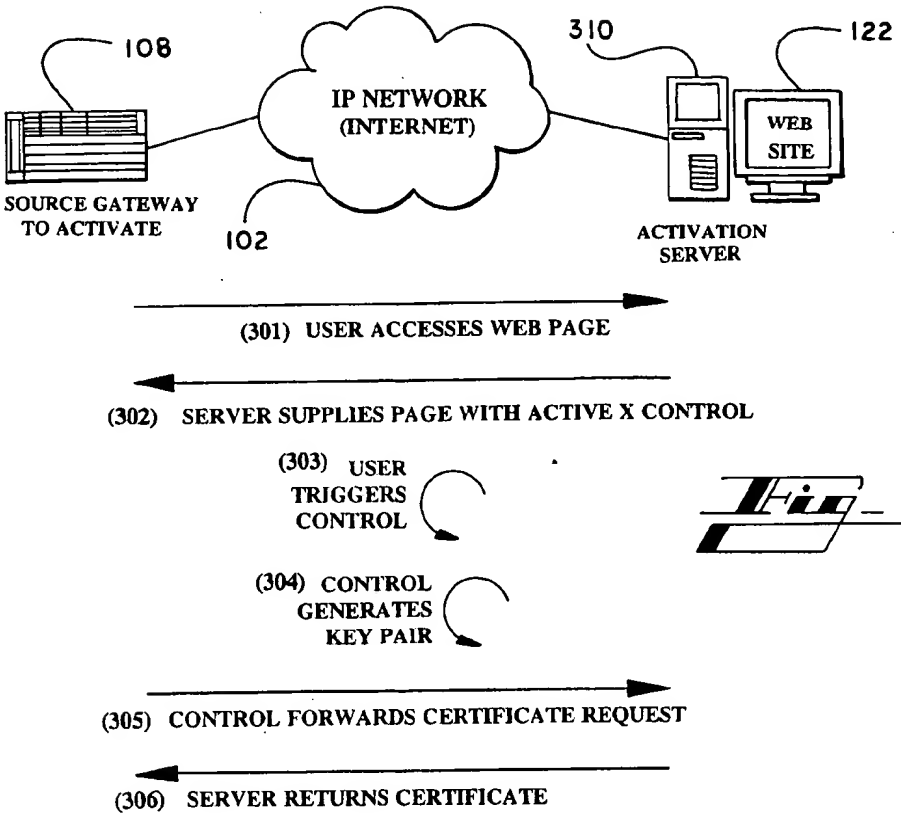
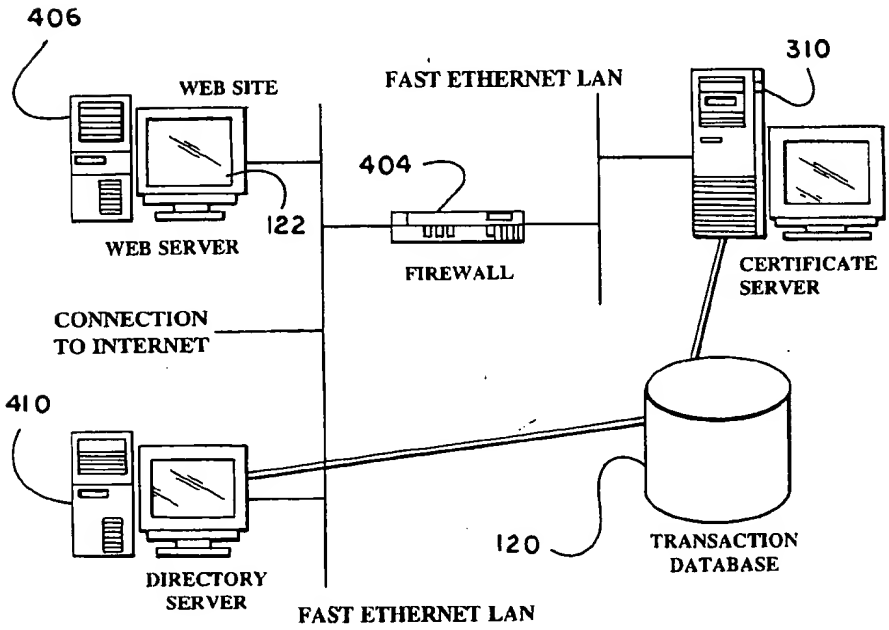

108
IP NETWORK
(INTERNET)
310
122
WEB
SITE
SOURCE GATEWAY
TO ACTIVATE
102
ACTIVATION
SERVER
(301) USER ACCESSES WEB PAGE
(302) SERVER SUPPLIES PAGE WITH ACTIVE X CONTROL
(303) USER
TRIGGERS
CONTROL
(304) CONTROL
GENERATES
KEY PAIR
(305) CONTROL FORWARDS CERTIFICATE REQUEST
(306) SERVER RETURNS CERTIFICATE

Fig_3

Fig_4


406
WEB SITE
FAST ETHERNET LAN
310
404
122
WEB SERVER
FIREWALL
CERTIFICATE
SERVER
CONNECTION
TO INTERNET
410
120
TRANSACTION
DATABASE
DIRECTORY
SERVER
FAST ETHERNET LAN

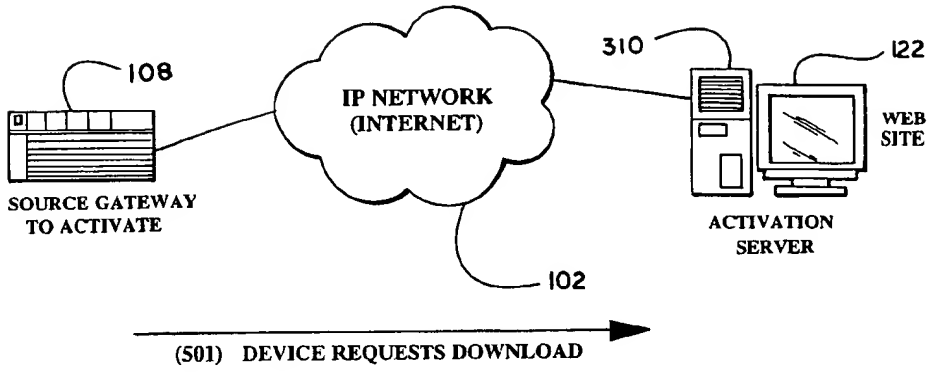
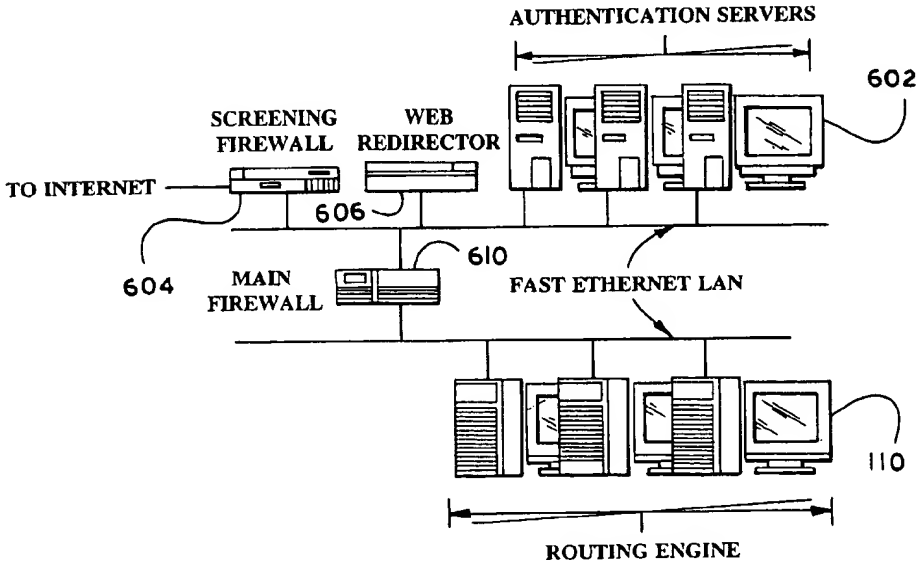

108
IP NETWORK
(INTERNET)
310
122
WEB
SITE
SOURCE GATEWAY
TO ACTIVATE
ACTIVATION
SERVER
102
(501) DEVICE REQUESTS DOWNLOAD
(502)
SERVER ADDS
ID TO BINARY
PROGRAM
(503)
SERVER
DIGITALLY SIGNS
BINARY PROGRAM
(504) SERVER DISTRIBUTES ACTIVATION PROGRAM

Fig. 5

Fig. 6


AUTHENTICATION SERVERS
602
SCREENING
FIREWALL
WEB
REDIRECTOR
TO INTERNET
606
610
604
MAIN
FIREWALL
FAST ETHERNET LAN
110
ROUTING ENGINE

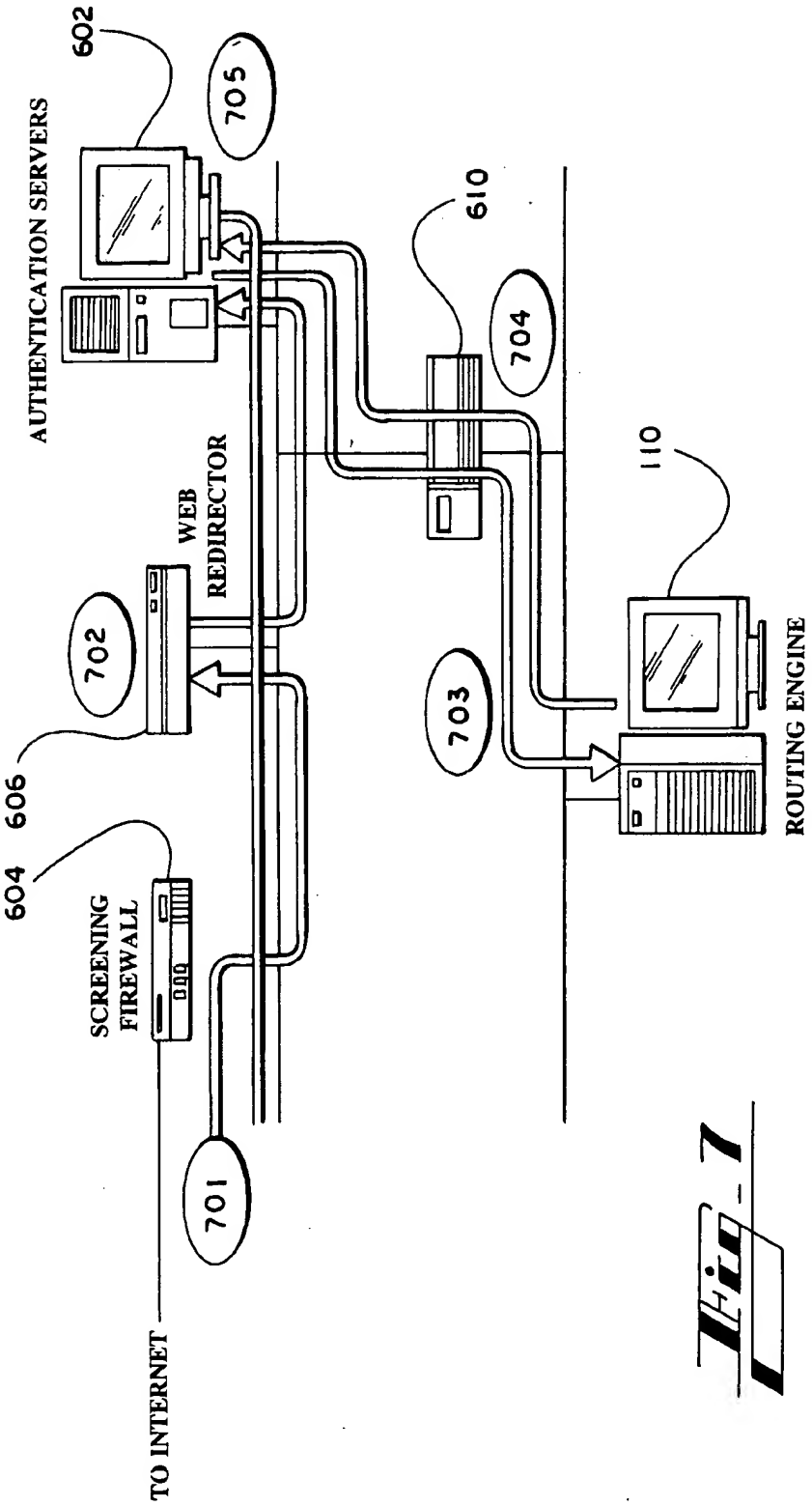
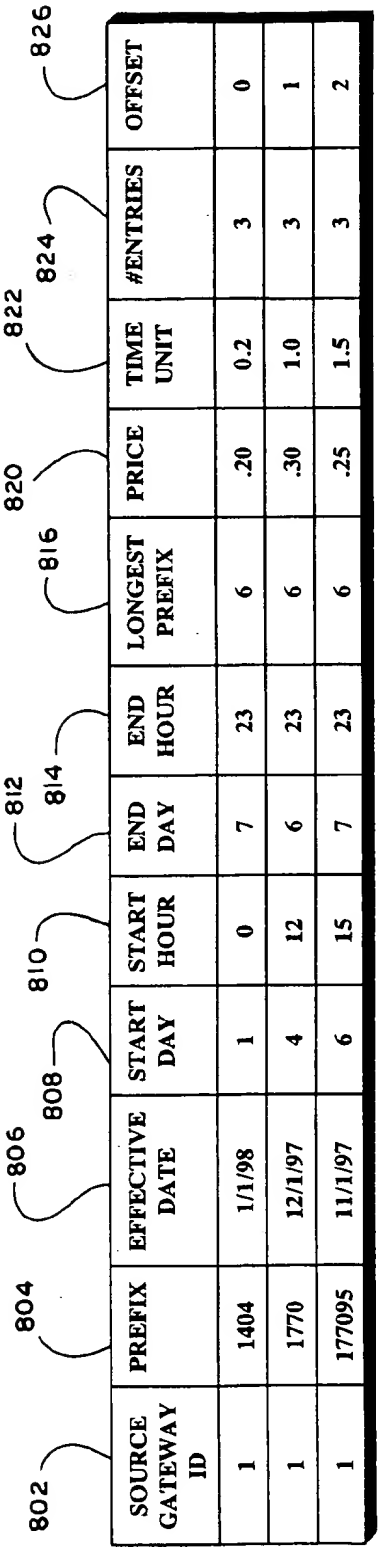

602
705
AUTHENTICATION SERVERS
610
704
110
WEB
REDIRECTOR
702
703
606
604
ROUTING ENGINE
SCREENING
FIREWALL
701
TO INTERNET

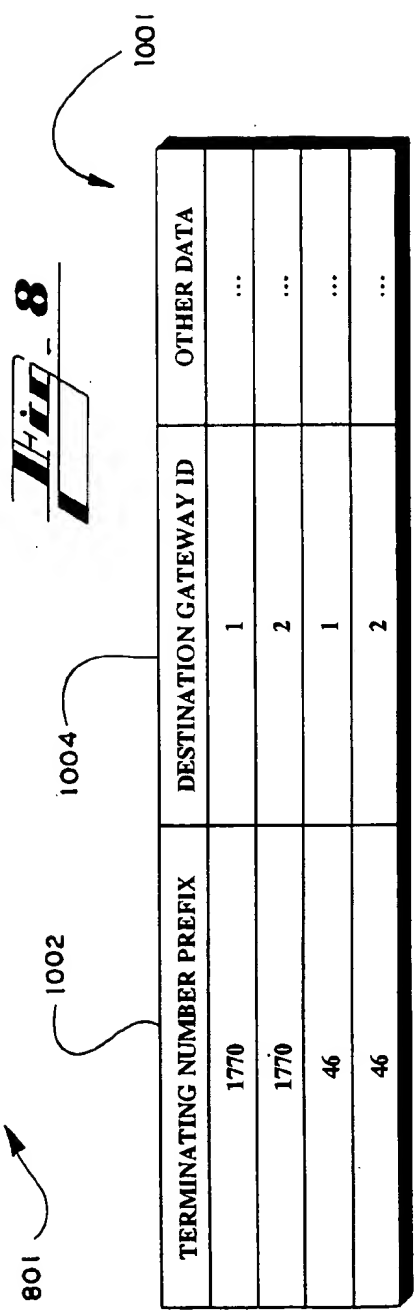
Fig. 7



SOURCE GATEWAY ID	PREFIX	EFFECTIVE DATE	START DAY	START HOUR	END DAY	END HOUR	LONGEST PREFIX	PRICE	TIME UNIT	#ENTRIES	OFFSET
1	1404	1/1/98	1	0	7	23	6	.20	0.2	3	0
1	1770	12/1/97	4	12	6	23	6	.30	1.0	3	1
1	177095	11/1/97	6	15	7	23	6	.25	1.5	3	2

802 804 806 808 810 812 814 816 820 822 824 826

Fig. 8



TERMINATING NUMBER PREFIX	DESTINATION GATEWAY ID	OTHER DATA
1770	1	...
1770	2	...
46	1	...
46	2	...

801 1001 1002 1004

Fig. 10

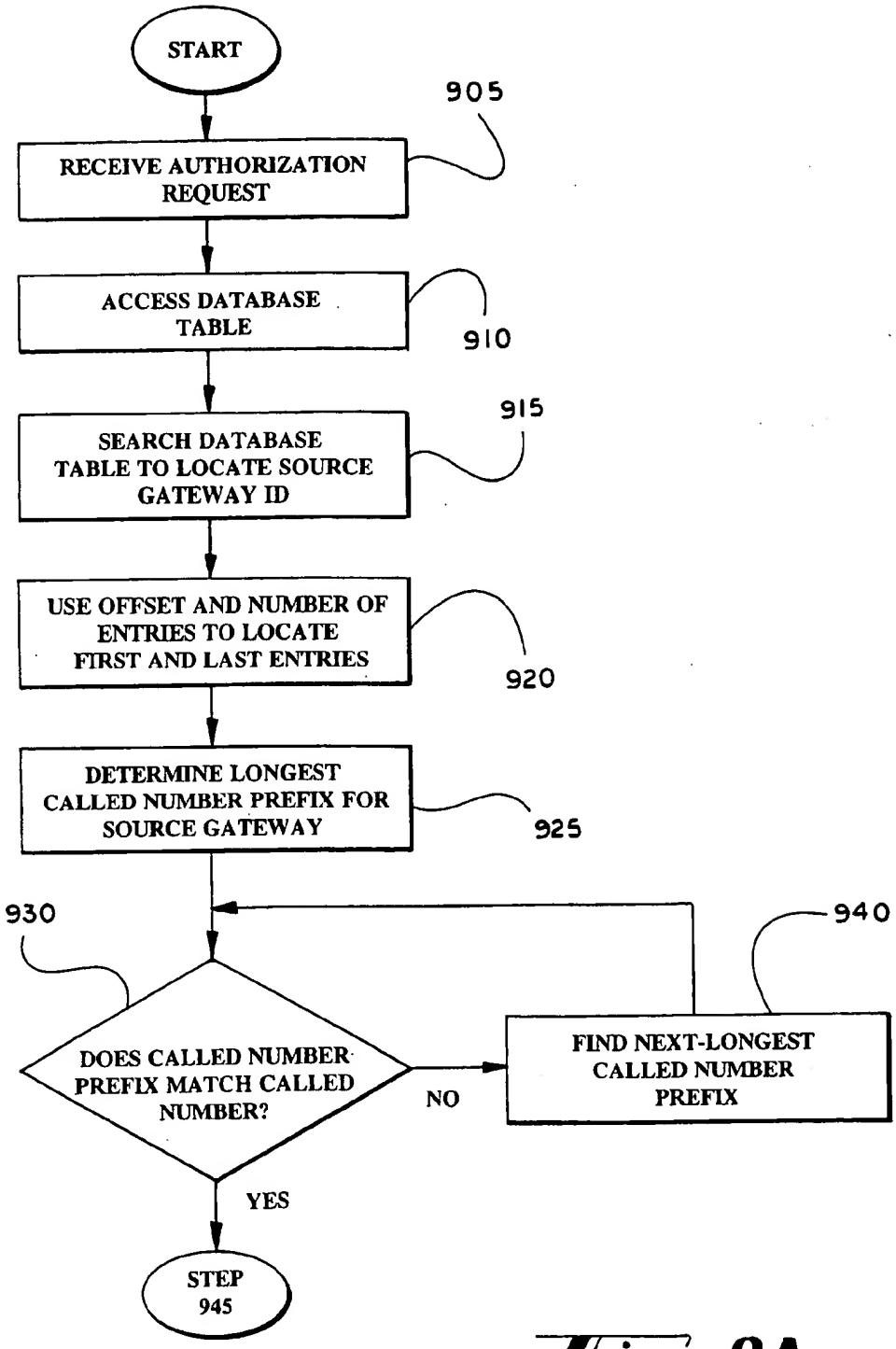

START
905
RECEIVE AUTHORIZATION REQUEST
ACCESS DATABASE TABLE
910
915
SEARCH DATABASE TABLE TO LOCATE SOURCE GATEWAY ID
USE OFFSET AND NUMBER OF ENTRIES TO LOCATE FIRST AND LAST ENTRIES
920
DETERMINE LONGEST CALLED NUMBER PREFIX FOR SOURCE GATEWAY
925
930
940
DOES CALLED NUMBER PREFIX MATCH CALLED NUMBER?
NO
FIND NEXT-LONGEST CALLED NUMBER PREFIX
YES
STEP 945

Fig. 9A

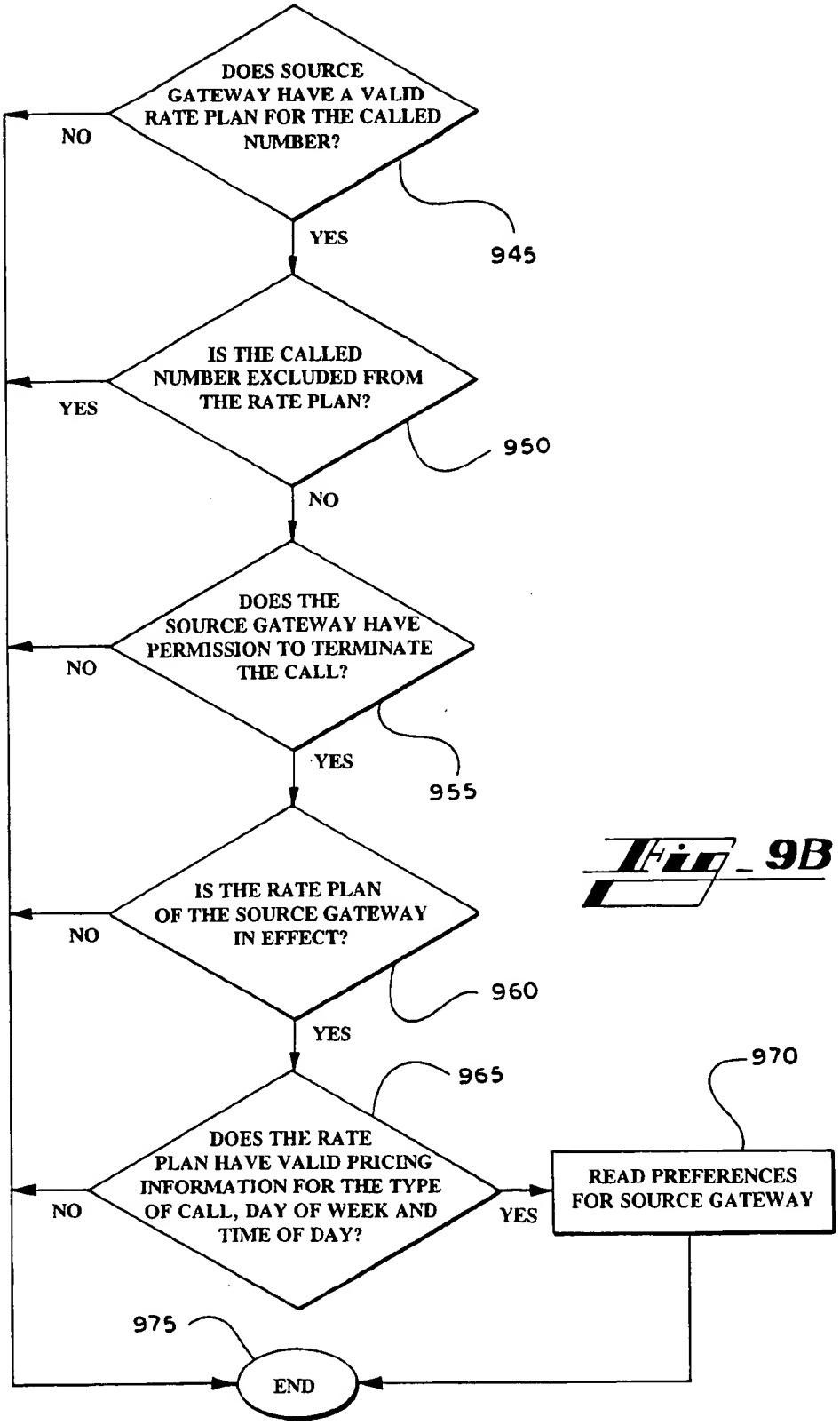

DOES SOURCE GATEWAY HAVE A VALID RATE PLAN FOR THE CALLED NUMBER?
NO
YES
945
IS THE CALLED NUMBER EXCLUDED FROM THE RATE PLAN?
YES
950
NO
DOES THE SOURCE GATEWAY HAVE PERMISSION TO TERMINATE THE CALL?
NO
YES
955
IS THE RATE PLAN OF THE SOURCE GATEWAY IN EFFECT?
NO
960
YES
965
DOES THE RATE PLAN HAVE VALID PRICING INFORMATION FOR THE TYPE OF CALL, DAY OF WEEK AND TIME OF DAY?
NO
YES
970
READ PREFERENCES FOR SOURCE GATEWAY
975
END

Fig. 9B

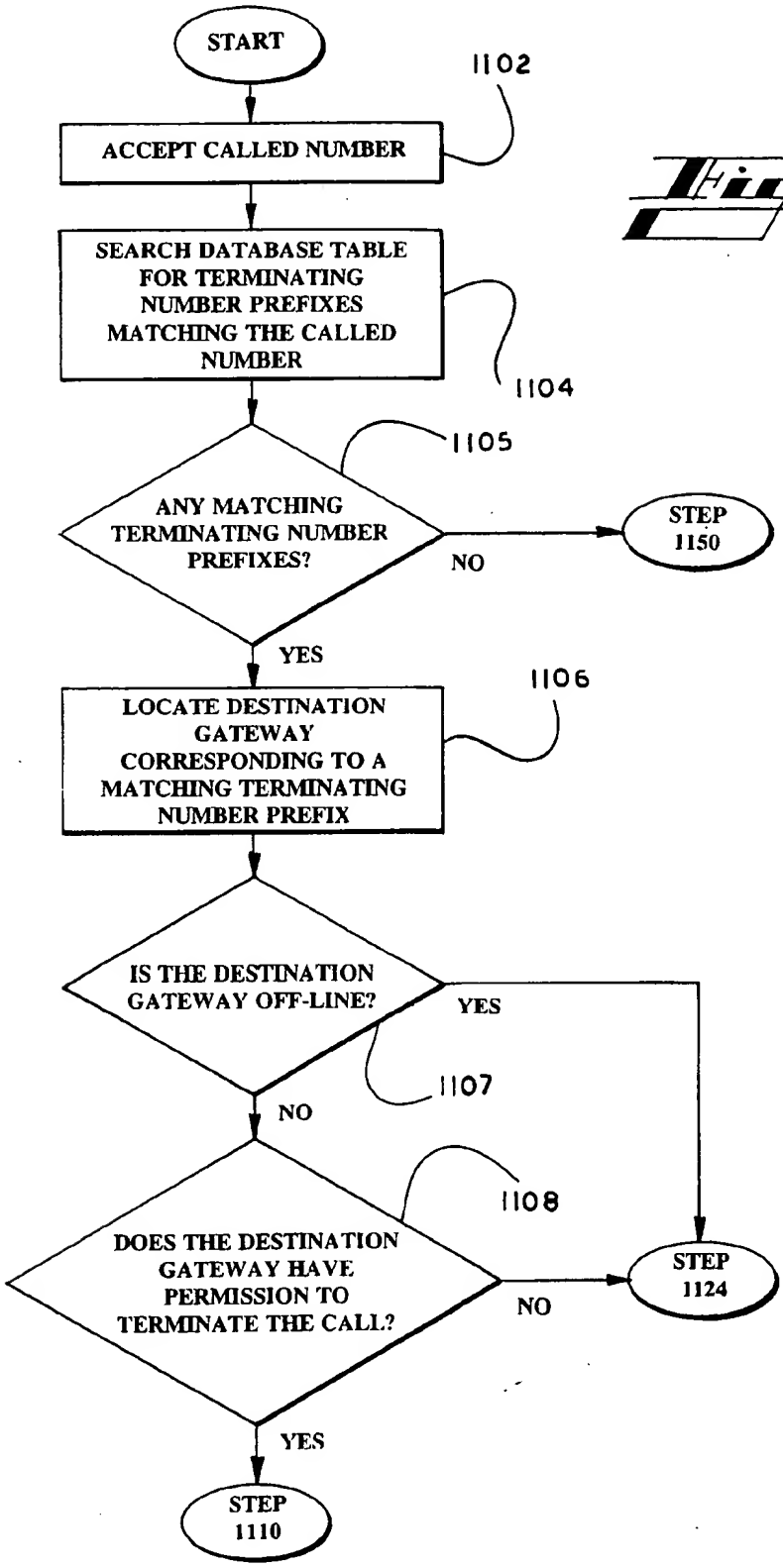

START
1102
ACCEPT CALLED NUMBER
SEARCH DATABASE TABLE FOR TERMINATING NUMBER PREFIXES MATCHING THE CALLED NUMBER
1104
1105
ANY MATCHING TERMINATING NUMBER PREFIXES?
NO
STEP 1150
YES
1106
LOCATE DESTINATION GATEWAY CORRESPONDING TO A MATCHING TERMINATING NUMBER PREFIX
IS THE DESTINATION GATEWAY OFF-LINE?
YES
1107
NO
1108
DOES THE DESTINATION GATEWAY HAVE PERMISSION TO TERMINATE THE CALL?
NO
STEP 1124
YES
STEP 1110

Fig. 11A

Fig. 11B

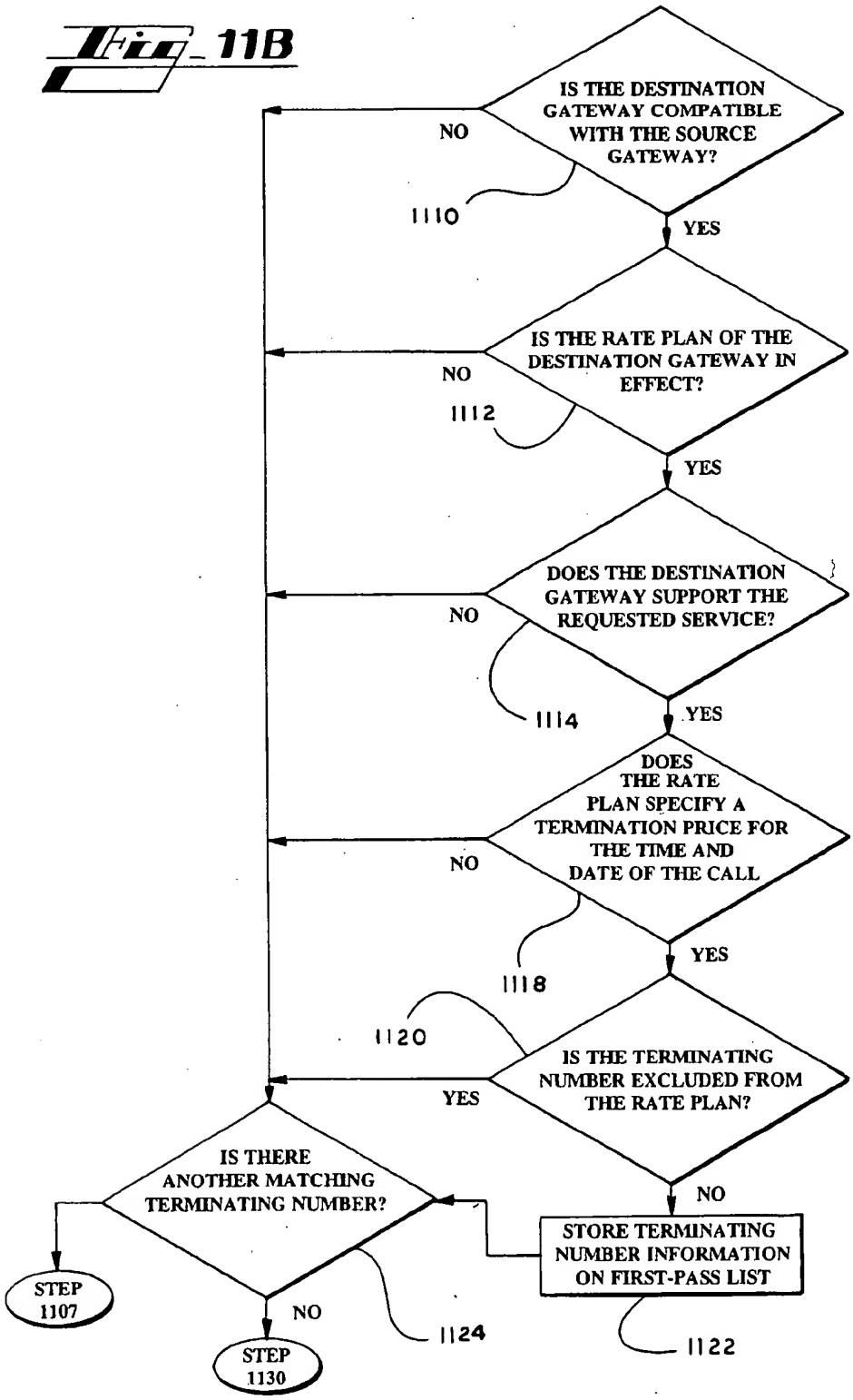
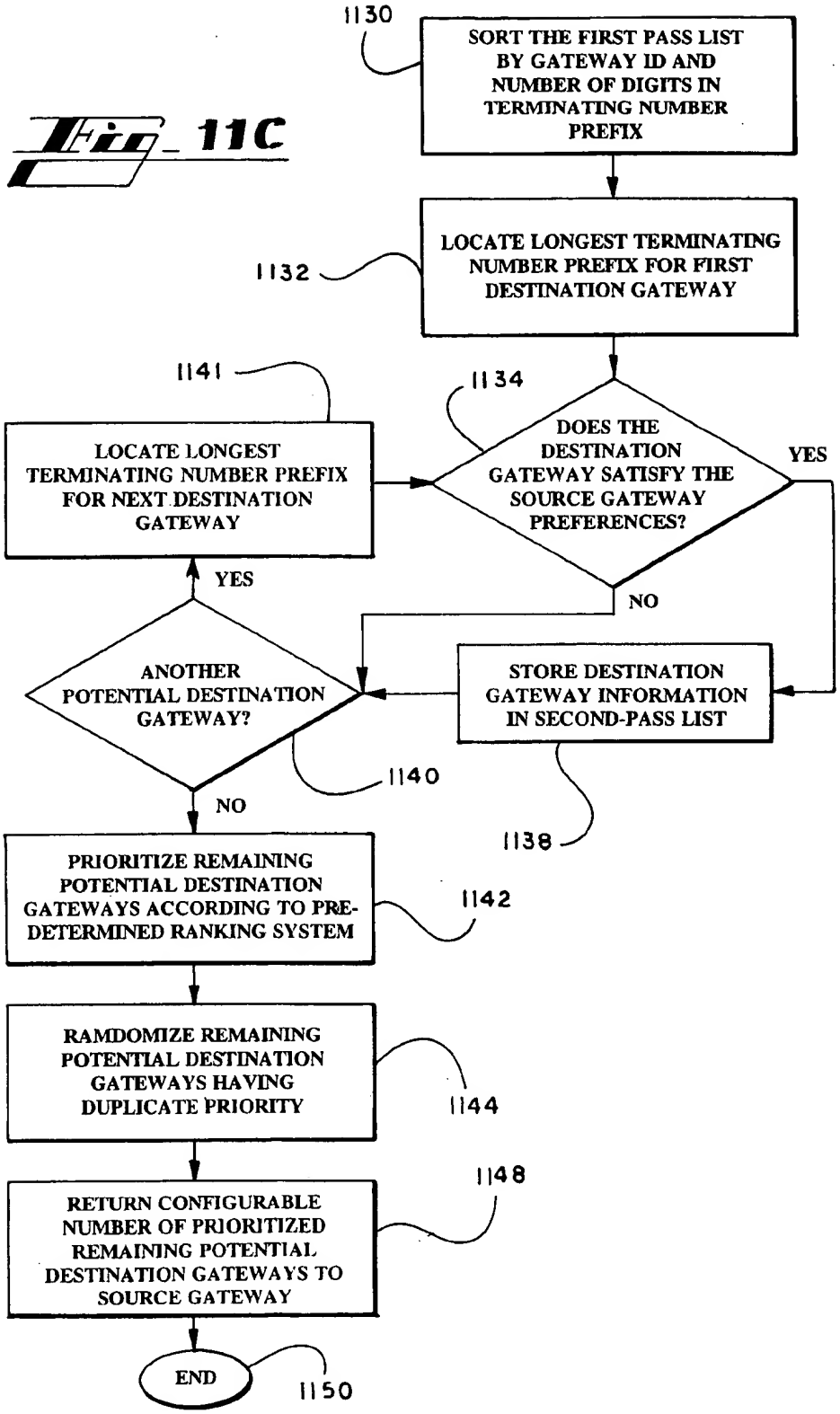

IS THE DESTINATION GATEWAY COMPATIBLE WITH THE SOURCE GATEWAY?
NO
1110
YES
IS THE RATE PLAN OF THE DESTINATION GATEWAY IN EFFECT?
NO
1112
YES
DOES THE DESTINATION GATEWAY SUPPORT THE REQUESTED SERVICE?
NO
1114
YES
DOES THE RATE PLAN SPECIFY A TERMINATION PRICE FOR THE TIME AND DATE OF THE CALL
NO
1118
YES
1120
IS THE TERMINATING NUMBER EXCLUDED FROM THE RATE PLAN?
YES
NO
IS THERE ANOTHER MATCHING TERMINATING NUMBER?
STEP 1107
NO
STEP 1130
1124
STORE TERMINATING NUMBER INFORMATION ON FIRST-PASS LIST
1122

Fig. 11C


1130
SORT THE FIRST PASS LIST BY GATEWAY ID AND NUMBER OF DIGITS IN TERMINATING NUMBER PREFIX
1132
LOCATE LONGEST TERMINATING NUMBER PREFIX FOR FIRST DESTINATION GATEWAY
1134
DOES THE DESTINATION GATEWAY SATISFY THE SOURCE GATEWAY PREFERENCES?
YES
NO
1141
LOCATE LONGEST TERMINATING NUMBER PREFIX FOR NEXT DESTINATION GATEWAY
YES
ANOTHER POTENTIAL DESTINATION GATEWAY?
1140
NO
STORE DESTINATION GATEWAY INFORMATION IN SECOND-PASS LIST
1138
PRIORITIZE REMAINING POTENTIAL DESTINATION GATEWAYS ACCORDING TO PRE-DETERMINED RANKING SYSTEM
1142
RAMDOMIZE REMAINING POTENTIAL DESTINATION GATEWAYS HAVING DUPLICATE PRIORITY
1144
1148
RETURN CONFIGURABLE NUMBER OF PRIORITIZED REMAINING POTENTIAL DESTINATION GATEWAYS TO SOURCE GATEWAY
END
1150

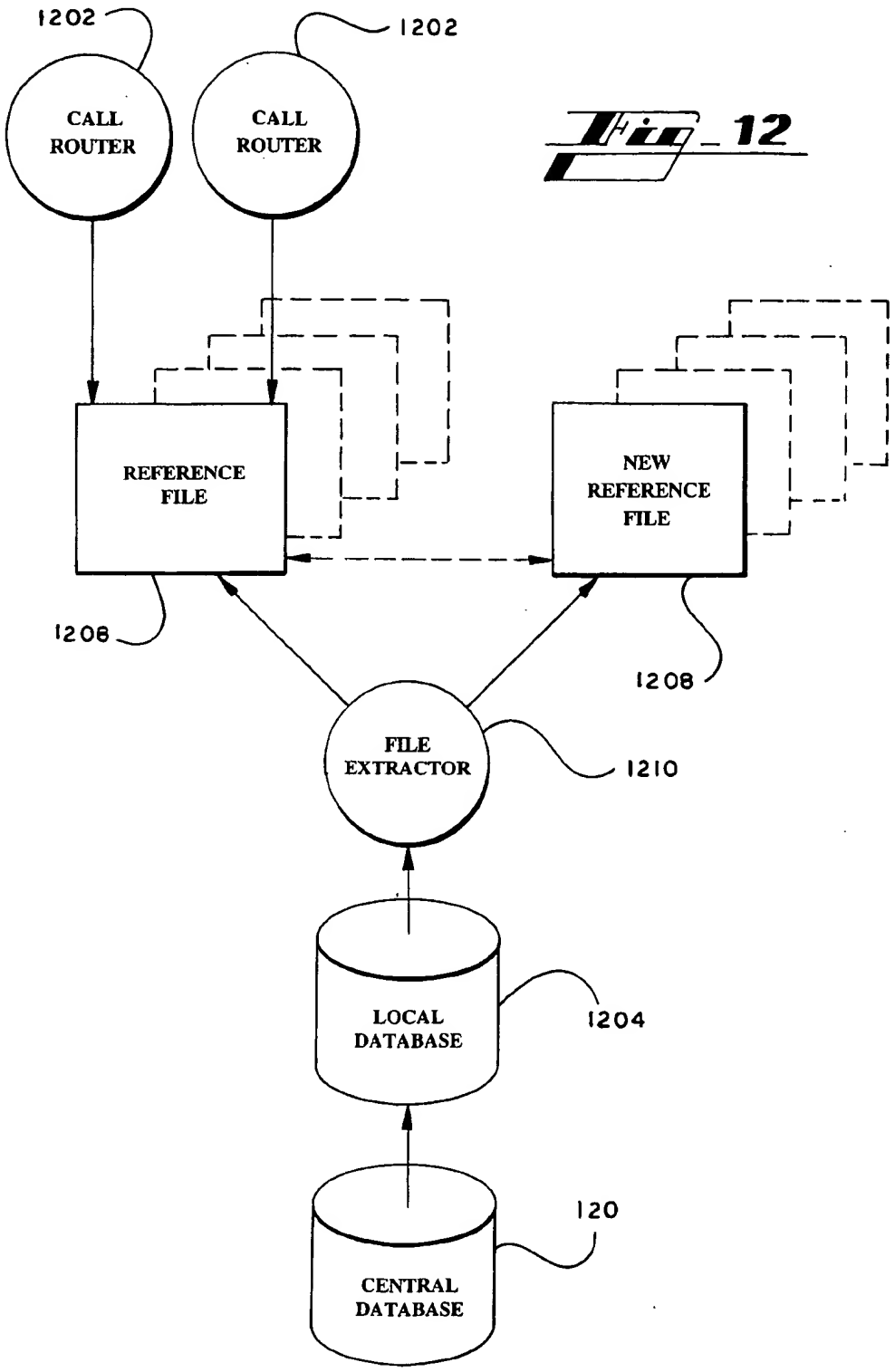

1202
1202
CALL ROUTER
CALL ROUTER
REFERENCE FILE
NEW REFERENCE FILE
1208
1208
FILE EXTRACTOR
1210
LOCAL DATABASE
1204
120
CENTRAL DATABASE

Fig. 12

INTERNET TELEPHONY CALL ROUTING ENGINE

RELATED APPLICATIONS

The present application claims priority to provisional patent application entitled "Internet Communications Clearinghouse System", filed on Sep. 16, 1997 and assigned U.S. application Ser. No. 60/059,087, and is related to application entitled "Gatekeeper for Internet Clearinghouse Communications System" filed on Sep. 16, 1998 and assigned U.S. application Ser. No. 09/154,566, now abandoned.

TECHNICAL FIELD

The present invention generally relates to voice over IP communications. More particularly, the present invention relates to a routing engine to assist in the routing of voice over IP communications from a source gateway to a destination gateway.

BACKGROUND OF THE INVENTION

As an alternative to traditional switched circuit networks, telecommunications service providers have discovered that voice telephone calls may be routed over IP networks. Due to the fact that the Internet is not presently subject to the same international regulations as are traditional telephone networks, routing telephone calls over the Internet tends to be less expensive. Additionally, an IP routed voice telephone call requires much less bandwidth, and thus less cost, than a voice telephone call placed over a traditional telephone network. Further, IP technology advances and is entered into the marketplace at a much faster rate than traditional telecom technology. Thus, in order to be competitive, telecommunications service providers have begun to use IP routing as a way to offer customers access to the latest technological improvements.

Presently, however, there is no centralized system for routing voice telephone calls over an IP network. Each operator of a gateway is responsible for determining the routes for its own outgoing calls. Typically, gateway operators rely on traditional IP routing algorithms, which are designed to handle routing of computer generated data packets. Traditional IP routing algorithms attempt to strike a balance between the concerns of minimum delay and maximum reliability. Thus, using traditional IP routing algorithms, a voice telephone call will be routed to any destination gateway that happens to satisfy a set of predetermined shortest path and acceptable data loss parameters.

The routing of voice telephone calls, however, involves a significant concern that is not shared by traditional IP routing algorithms. This additional concern is the monetary cost of routing a voice call to a particular destination gateway. As in traditional switched circuit networks, Internet telephony gateways impose fees for the service of terminating a voice call. Traditional IP routing algorithms are not able to detect and compare the varying price schedules that may be imposed by various Internet telephony gateways. Thus, source gateways are not able to discriminate between destination gateways based on monetary costs.

Thus, there remains a need in the art for voice over IP routing that is able to balance financial concerns with concerns for minimum delay and maximum reliability.

There also remains a need in the art for a centralized system for assisting gateway operators with routing decisions.

SUMMARY OF THE INVENTION

The present invention relates to a routing engine connected to an IP network, such as the Internet, that provides

gateways with assistance in the routing and billing of voice over IP transactions. The novel routing engine provides a source gateway with a prioritized list of destination gateways that are eligible to terminate a voice telephone call. The routing engine locates eligible destination gateways by gathering and matching information relating to "preferences" from various gateway operators. For a source gateway, preferences may be the maximum price that will be paid for a given call, the maximum delay that will be tolerated for the call and the maximum autonomous system hop count that will be tolerated. For a destination gateway operator, the most relevant preference is the price charged for access to the destination gateway.

Gateway operators may also designate "preference criteria," which define the circumstances in which a given set of preferences are to apply. Preference criteria may relate to the identification of a particular gateway, a particular called number prefix, a particular time of day and/or day of the week. Thus, for example, a source gateway operator may specify that all calls place from a particular source gateway will only tolerate a stated amount of delay and will only incur a set amount of costs. Also, a destination gateway operator may specify that a certain price will be charged for access to a certain gateway at a certain time of day, or for calls placed to a specific geographic region, or even for calls placed to a specific telephone number. Routing, and thus billing, flexibility is virtually limitless due to the designation of preferences and preference criteria.

Gateway operators designate preferences and preference criteria through a web-site that is related to the routing engine, or through other electronic transfer means. The preferences and preference criteria are then transferred to a centralized database that is accessible to all routing engines that may be distributed around an IP network. Geographically distributed routing engines are desirable in order to handle requests for routing assistance from geographically diverse gateways. Additionally, at a given location, a scalable number of routing engines may be coupled together, so as to process a multitude of routing requests with speed and efficiency.

Thus, it is an object of the present invention to provide routing that is able to account for financial concerns as well as signal delay and quality of communications service. It is a further object of the present invention to provide a centralized service point to assist gateways in the process of making voice over IP routing decisions.

These and yet other objects, features and advantages of the present invention will become apparent from reading the following specification, taken in conjunction with the accompanying drawing.

BRIEF DESCRIPTION OF THE DRAWING

FIG. **1** is a schematic representation of an exemplary operating environment for the present invention;

FIG. **2** provides an overview of the steps involved in an Internet telephony call in the exemplary operating environment;

FIG. **3** provides an example of the device activation process for a device running the Win32 platforms;

FIG. **4** provides a detailed picture of an exemplary activation server as a part of the exemplary operating environment;

FIG. **5** illustrates the steps involved in the activation of a UNIX-based source gateway;

FIG. **6** shows the general architecture of a service point;

FIG. 7 illustrates the overall message flow within a service point;

FIG. 8 illustrates an exemplary database table for storing information relating to a source gateway;

FIG. 9A describes an exemplary method by which a routing engine may access a database table to locate preferences for a source gateway;

FIG. 9B is a continuation of FIG. 9A;

FIG. 10 illustrates an exemplary database table for storing information relating to a destination gateway;

FIG. 11A shows an exemplary method that may be used by a routing engine to locate eligible gateways in a database table;

FIG. 11B is a continuation of FIG. 11A;

FIG. 11C is a continuation of FIG. 11B;

FIG. 12 describes the internal architecture of an exemplary routing engine.

DETAILED DESCRIPTION OF THE EXEMPLARY EMBODIMENTS

The present invention relates to a routing engine for routing telephony calls from a source gateway to a destination gateway via an IP network. A telephone call occurring via an IP network is often referred to as a "voice over IP" transaction. When a "voice over IP" transaction specifically involves the Internet, the description "Internet telephony" may also used to describe the transaction. An exemplary embodiment of the routing server will be described with respect to Internet telephony. However, the principles of the routing engine of the present invention apply to all IP routed transactions, including, but not limited to, "voice over IP" calls, "fax over IP" calls, and "video over IP calls."

Exemplary Operating Environment

The following description of an exemplary operating environment and exemplary embodiments of the present invention will refer to the drawing, in which like numerals indicate like parts throughout the several figures. Referring thereto, FIG. 1 shows an network architecture that serves as an exemplary operating environment for the routing engine of the present invention. As indicated, the Internet **102** serves as the heart of the exemplary network architecture. Relying on the Internet **102** are five different systems that might participate in an Internet Telephony transaction. These five systems include: a calling party **104**, a source gateway (also referred to as an originating gateway) **108**, a service point **112** including a routing engine **110**, a destination gateway (also referred to as a terminating gateway) **114** and a called party **118**. As FIG. **1** shows, a service point **112** is coupled to a central database **120**, which is also coupled to a billing and settlement system **124**. While the service point **112** exists on the public Internet **102**, the central database **120** and the billing and settlement system **124** remain in secured facilities. Private communication paths connect the remote equipment with the central database **120**.

The calling party **104** represents the user wishing to place a telephone call. Often, the calling party **104** will rely on a standard telephone handset to place the call. In fact, in many cases the calling party **104** may not be able to distinguish Internet telephony service from standard telephone service. The calling party **104** connects to a source gateway **108** through a public telephone network **105**, such as a switched circuit network. In either case, the source gateway **108** serves as a bridge between ordinary telephones and the Internet **102** by converting telephone signals into data packets (and vice versa) and transmitting the data packets over the Internet **102**. A source gateway is operated by a source gateway operator **109**.

Similarly, the called party **118** is the user that receives a telephone call. A called party **118** connects to a destination gateways **114** through a public telephone network **106**, such as a switched circuit network. A destination gateway **114** is connected to the Internet **102** at a location that is remote from the source gateway **108**. The destination gateway **114** is operated by a destination gateway operator **115** and performs the same functions as the source gateway **108**, i.e., bridging phone calls between the Internet **102** and a public telephone network **106**, or an equivalent thereof. Destination gateways **114** differ from source gateways **108** only in the role played in a particular call. In particular, source gateways **108** act on behalf of the calling party **104**, while destination gateways **114** act on behalf of the called party **118**. It is important to note that the same operator need not manage both the source gateway **108** and the destination gateway **114**. In fact, the exemplary routing engine **110**, is tailored for environments in which different owners operate the two types of gateways.

The service point operator **125** may be a third party that is independent of the operators of the source gateway **108** or destination gateways **114**. As indicated in FIG. **1**, the service point operator may maintain a private communications line with the service point **112**, the billing and settlement system **124** and a related web-site **122**. In the exemplary operating environment, all components maintained by the service point operator **125**, i.e., the service point **112**, the database **120**, the billing and settlement system **124** and the web-site **122**, are conveniently distributed between various geographic locations. Still, those skilled in art will appreciate that all components maintained by the service point operator **125** may be incorporated in a single system (service point **112**) or any number of distributed systems.

A service point **112** communicates with gateways over the Internet **102** and generally provides routing information to the source gateway **108**. Given a destination phone number and other requirements (described in detail below), the service point **112**, through the routing engine **110**, identifies at least one appropriate destination gateway **114** to handle the telephone call.

The overall network architecture that serves as an operating environment for the exemplary routing engine **110** may be thought of as comprising three different networks, each carrying the telephone conversation. The first network is the calling party's telephone network **105** that connects the calling party to the source gateway **108**. The second network is the Internet **102**, which connects the source gateway **108** and the destination gateways **114** to each other. The third network is the called party's telephone network **106**, which completes the connection from the destination gateway **114** to the called party **118**. Although FIG. **1** (as well as this description in general) refers to the telephone connections as taking place through public telephone networks **105** and **106**, Internet telephony service does not require such a connection. Some applications may use private networks, such as those provided by a private branch exchange; others may simply connect telephone handsets directly to the corresponding gateway.

Additionally, a fourth network may be added to the general network architecture. The fourth network is a banking and funds transfer network **126**. A billing and settlement system **124** may be coupled to the service point **112** in order to receive information relating to the financial aspects of the Internet telephony transactions. The billing and settlement system **124** may use a banking and funds transfer network

126 to execute the financial transactions coordinated by the service point 112.

FIG. 2 provides an overview of an Internet telephony call in the exemplary operating environment. At step 201, an Internet telephony call is initiated when the calling party 104 dials a telephone number, which is transmitted to the source gateway 108 for processing. The goal of the source gateway 108 is to locate a destination gateway 114*a–c* that is able to terminate the phone call. The source gateway 108 relies on the service point 112 for routing assistance.

At step 202, the source gateway 108 makes an authorization request to a service point 112. The authorization request indicates, among other things, the telephone number of the called party 118. At the service point 112, the routing engine 110 uses information in the authorization request, as well as preferences established for the source gateway's 108 cost and quality requirements, to determine which of the destination gateways 114*a–c* are eligible to complete the call.

At step 203, the service point 112 then sends an authorization response message to the source gateway 108, which includes information relating to the identity of eligible destination gateways 114. In addition, the authorization response message contains an authorization ticket for access to each eligible destination gateway 114. The authorization response ticket allows a destination gateway 114 to accept the call knowing that it has been authorized by the service point 112, and that the service point operator 125 will compensate the destination gateway operator 115 for completing the call.

Upon receipt of the authorization response message, the source gateway 108 selects a destination gateway 114 from among the list provided by the service point 112. At step 204, the originating gateway 108 then sends a setup message to the selected destination gateway 114, as specified in International Telecommunications Union (ITU) H.323 and associated standards. Those skilled in the art will recognize that the Q.931 standard may be used to define the setup message. To complete the authorization, the setup message must include the authorization ticket for the destination gateway 114. Those skilled in the art will also recognize that the user-to-user information element of the Q.931 setup message may be used to convey the authorization ticket.

Communication between the service point 112, the source gateway 108 and the destination gateways 114 does not require the use of standard protocols for any aspect of the Internet telephony calls themselves, including call setup. If the source gateway 108 and destination gateways 114 use a signaling protocol other than Q.931 (which is specified by H.323 and H.225.0), then that protocol need only be capable of including the authorization ticket in the initial setup message. The exemplary authorization ticket is approximately 2000 octets in length. Destination gateways 114*a–c* may accept or reject Internet telephony calls based on the presence and contents of this authorization ticket.

After the Internet telephony call is completed, both the source gateway 108 and the destination gateway 114 transmit a call detail report to the service point 112, as represented in steps 205 and 206. Call detail reports identify the call and record its duration. Call detail reports are stored in the database 120 and are accessed by the billing and settlement system 124 in order to reconcile financial obligations between the service point operator 125, source gateway operators 109 and destination gateway operators 115.

It should be noted that source gateway 108 and destination gateways 114 are free to establish connections without consulting a service point 112. For example, a group of gateways may all be owned by a common entity and may wish to exchange calls among themselves independent of a service point 112. In such an environment, the gateways are free to rely on a service point 112 only when no gateway in the group can serve a given phone number economically. Thus, the exemplary operating environment provides gateways with extremely flexible routing choices.

Also, those skilled in the art will appreciate that the exemplary operating environment may include multiple service points 112. Service points may be distinguished by the specific services they provide, as well as by their geographic location on the Internet 102. Geographic diversity optimizes performance by allowing a device to communicate with the closest service point 112. Proximity to a service point 112 minimizes delay in the communication exchange. Geographic diversity also increases the reliability of the operating environment. If one service point 112 becomes unavailable, devices using that service point 112 can automatically switch to a different service point (not shown) located elsewhere.

Before a gateway is provided with access to a service point 112 the responsible gateway operator must enroll as a customer of the service point operator 125. The customer enrollment process may take place through the web-site 122, via the Internet 102, using any well-known web browser. Gateway operators 109 & 115 typically perform the enrollment from a desktop computer. Since the enrollment process typically requires disclosure of sensitive financial information (such as bank accounts or credit card numbers), the web connection between the gateway operators 109 & 115 and the web-site 122 is secured by the secure sockets layer (SSL) protocol. The web-site 122 uses SSL to authenticate itself to gateway operators 109 & 115 with digital certificates obtained from a trusted certificate authority. SSL also encrypts the information transferred between the gateway operators 109 & 115 and the web-site 122.

When the service point operator 125 accepts a gateway operator as a customer, it provides the customer with a customer number and password. The customer number is Hamming coded to protect against corruption. Once assigned, customers are allowed to change their password. The service point operator 125 may enforce certain restrictions on passwords to maximize security. Such restrictions may include, for example, a prohibition against words appearing in dictionaries, a requirement to use both upper and lower case characters and a requirement that customers change their password periodically.

After enrollment is complete, gateway operators 109 & 115 are given authorization to access and modify their accounts, via the Internet 102, through the web-site 122. Enrolled customers may also be provided with access to timely and informative reports on their usage of a service point 112. Such reports may include up-to-the-minute billing information, potential fraud alerts, sophisticated usage statistics and detailed traffic profiles. Enrolled users may access these reports directly through the web-site 122, using a web browser, or they can download the information for importing into their own database or spreadsheet. Users may also elect to be notified via electronic mail, fax, or other means when certain events occur. Events eligible for this service include suspicious or fraudulent activity, minimum or maximum traffic levels at particular devices, and apparent failure of a device.

An enrolled customer may activate individual devices to use the services provided by a service point 112. In the present discussion, the exemplary devices are Internet telephony gateways 108 & 114. However, those skilled in the art

will appreciate that the exemplary operating environment may be configured to supports a wide variety of devices. As with operator enrollment, device activation takes place across the Internet **102** using well-known web browsers. Typically, device activation will take place at the device itself, while operator enrollment is performed from an operator's personal computer or workstation.

A web-site **122** may be configured to support several different approaches for activating devices, depending on the particular type of device. In all cases, though, a device becomes activate through a three-step process. First the device generates a public/private key pair and stores the private key securely. Next the device forwards the public key to the web-site **122** through a certificate request. Last, the web-site **122** provides a certificate to the device validating the public key. The detailed implementation of this three-step device activation process varies depending on the operating environment of a particular device. A web-site **122** may be configured to support Windows, UNIX, and embedded operating environments. Those skilled in the art will recognize that other operating systems may also be supported.

With respect to the Windows operating environment, exemplary web-site **122** may be designed to support the operating environments of Windows 95, Windows 98 and Windows NT version 4.0 and later (collectively referred to as "Win32 platforms"). For these operating environments, reliance may be placed heavily on Microsoft's Internet Explorer (version 3.02 and later) to generate key pairs and to request and install certificates. The Certificate Server component of Microsoft's Internet Information Server version 4.0 may be used to grant certificate requests.

FIG. **3**, shows an example of the activation process for a device running the Win32 platforms. The two systems involved in the communication are the device being activated, i.e. source gateway **108**, and an activation server **310** (also known as a certificate server) running as part of the web site **122**. As indicated at step **301**, the user of the source gateway **108** must first navigate to the appropriate part of the web site **122**, such as a device activation page. Users may be prompted to provide authentication information to access a device activation page. In addition to HTML-formatted instructions, the device activation page downloads an ActiveX control, as shown as step **302**. This control is digitally-signed by a trusted object-signing authority. When triggered by appropriate user interaction with the HTML form at step **303**, the control causes the device to generate a public/private key pair at step **304**, build a certificate request and forward that request at step **305** to the activation server **310**.

The ActiveX control relies on the version 2.0 of the Crypto API interface, available on Win32 platforms, for cryptographic algorithms and certificate management. Optimally, the control will use the strongest cryptographic algorithms available on the platform. For example, if the source gateway **108** has installed Microsoft's enhanced cryptographic service provider, then the control will use its cryptographic services. In the absence of other services, the control will use Microsoft's base cryptographic service provider. If no cryptographic services are available in the device, device activation is not possible.

Certificate requests follow the format defined by RSA laboratories in Public Key Cryptography Standard (PKCS) number 10. At step **306**, the activation server grants the request and returns a certificate for the device (source gateway **108**). The certificate conforms to the International Telecommunication Union X.509 version 3 standard. Optimally, certificates issued by the activation server **310** will include the "subjectAltName" extension, marked as critical. That extension encodes both the customer and device number in ASCII text. The exemplary format for the data is "Customer=nnnnnnn, Device=mmmmmm" where nnnnnnn is the customer number and mmmmmm is the device number. Device numbers, like customer numbers, are generated by the service point operator **125** and are Hamming coded to protect against corruption.

Although shown in FIG. **3** as a single system, the activation server **310** may actually consist of multiple components. FIG. **4** provides a more detailed picture of an exemplary activation server **310**. Note that the actual activation server **310** is isolated from the Internet **102** by a firewall **404**. Web servers **406** outside the firewall provide the web-site **122**, but these web servers **406** must contact the activation server **310** through the firewall **404** to generate certificates. The activation server **310** stores copies of the certificates in a Open Database Connectivity (ODBC) accessible database **120**. The database **120** can redistribute certificates and certificate revocation lists (CRLs) to a service point **112** (where they are used to authenticate devices and authorize communications) as well as to public LDAP-compliant directories **410**. The public directories let customers of the service point operator **125** use certificates for additional security services such as gateway-to-gateway authentication and encryption.

UNIX-based devices cannot, in general, rely on the services of ActiveX controls and Internet Explorer for device activation. Instead, as shown in FIG. **5**, at step **501**, such devices must download binary software from the web site **122**. Such software may be optimized for several popular UNIX variants. To prevent unauthorized use of the software, the user's customer number, a generated device number, and a relatively short time limit (for example 15 minutes) are added to the software just prior to download at step **502**. The software is then digitally signed at step **503** and distributed to the Unix-based device (gateway) at step **504**. The precautions taken in the distribution of activation software effectively make the software a single-use program. Thus, the activation software cannot be re-distributed and used on other devices.

As an example, the activation software may include cryptographic software from RSA laboratories. In particular, the BSAFE 3.0 for cryptographic algorithms and BCERT 1.0 for certificate management software programs may be used. The activation server **310** may also be configured to automatically verify the physical location of a device with reverse Domain Name Service (DNS) and "who is" lookups. In cases where it is not possible to provide cryptographic software at all, for example due to import restrictions, device activation is not possible. Once downloaded, however, the activation program generates a key pair, formats a certificate request, and forwards that request to the activation server. The same activation server **310** may support both Windows and UNIX devices.

The web-site **122** may also be configured to support devices using embedded operating systems such as Cisco's Internetwork Operating System (IOS), WindRiver's VxWorks and others. To the extent that an embedded operating system can support standard UNIX services, it may use the approach outlined above for UNIX environments. Rather than supply downloadable binary programs for embedded environments, the web-site **122** may provide source code licenses to vendors using embedded environments. The method for embedding customer number and device number in that software may be defined on a case-by-case basis with each vendor.

Once enrolled customers have activated their devices (gateways), the devices can begin using the service to help complete their Internet telephony transactions. As mentioned, an Internet telephony transaction is initiated when a calling party **104** dials the telephone number of a called party **118**. The dialed telephone number is transmitted to the source gateway **108** for processing. The source gateway **108** must then locate a service point **112** that will provide routing assistance for the telephone call. As noted previously, several services points **112** may be connected to the Internet **102** to provided geographic diversity.

In the exemplary operating environment, service points **112** share a primary DNS name (such as "routing.transnexus.com.") Thus, source gateway **108** or other device may locate a service point **112** by simply attempting to communicate with the appropriately named system. Using DNS names allows for the use of technology such as Cisco's Distributed Director. When a source gateway **108** or other device requests a DNS lookup of a particular name, the Director automatically supplies the IP address of the service point **112** nearest the requesting device. By communicating with the nearest service point **112**, devices experience the minimum delay in accessing a service point **112**. In case the Distributed Director technology is unavailable, devices may also be configured with a list of specific names for individual service points **112**.

Specific names for individual service points may be of the form "us.routing.transnexus.com," "routing.transnexus.co.uk," and "routing.transnexus.cojp," where one component of the name indicates the service point's **112** location. Devices (gateways) should also be manually configured with their own current location, so that they can prioritize eligible service points **112** by proximity. A device can then try to contact each service point **112**, in turn, until communication is successful.

Once the source gateway **108** finds a service point **112**, it may access the services provided by the service point **112**. Service points **112** allow at least three forms of access. Hypertext Transfer Protocol (HTTP) is available for all types of service. Voice and fax services have two additional options, namely gatekeeper access and gatekeeper-routing. Service point access may be accomplished in the manner described in U.S. application Ser. No. 09/154,566, entitled "Gatekeeper for Internet Clearinghouse Communications System" filed on Sep. 16, 1998 and owned by the assignee for the present application. This related application, U.S. application Ser. No. 09/154,566, is hereby fully incorporated herein by reference.

When any device attempts to contact a service point **112**, using either HTTP or H.323 protocols, the service point **112** authenticates that device before providing service. Authentication relies on public key cryptography, most specifically the public/private key created during device activation, as described above. All messages from devices are digitally signed, using the device's private key. The message may also include a certificate validating the device's public key. A service point **112** obtains the device's public key, either from the included certificate (in which case it then verifies the certificate's signature), or directly from a certificate store. The public key permits verification of the signature.

The exemplary service point **112** architecture provides for flexible and scalable authentication services. As FIG. **6** shows, each service point **112** consists of a number of authentication servers **602**. The authentication servers **602** are protected by a screening firewall **604**, while a local redirector **606** provides load balancing and fault tolerance among the authentication servers **602**. All service points **112** preferably include at least two authentication servers **602** for fault tolerance, but can support many additional authentication servers **602** as load demands. FIG. **6** shows authentication servers **602** as standalone systems for clarity. However, those skilled in the art will recognize that actual implementation may involve rack-mounted components with a shared keyboard and monitor.

Authentication servers **602** may use the Windows NT operating system and the cryptographic services available in version 4.0 (SP3) and later. Authentication servers **602** are capable of software-based cryptographic services, but can be upgraded to hardware-based encryption technology as load demands. For devices that support multiple end users, such as Internet Telephony gateways **108** & **114**, authentication servers **602** may also be configured to support end-user level authentication. End-user identification and authentication (such as calling card and PIN numbers) may be included with each service request. Although optional, the end-user identification allows a service point **112** provide several enhanced services to its customers. Enhanced services may include sophisticated fraud control, end-user billing, and roaming services.

Once a service point **112** has authenticated a device (gateway), it can provide routing services for that device. In the exemplary operating environment, routing services may rely on special purpose routing engines **110**, which will be described in detail below. Since routing information is often sensitive data, routing engines **110** within a service point **112** are protected by an additional firewall **610**. As with authentication servers **602**, an exemplary service point **112** includes multiple routing engines **110** for scalability and fault tolerance. FIG. **6** shows how routing engines **110** connect to the service point **112** infrastructure. Again, routing engines shown as standalone systems for clarity may typically be implemented as rack-mounted components. Routing engines **110** preferably run routing software on high-performance UNIX servers. In the exemplary operating environment, each routing engine **110** operates autonomously, independent of other routing engines **110** in the service point **112**.

FIG. **7** illustrates the overall message flow within a service point **112**. At step **701**, an incoming authorization request message is filtered by the screening firewall **604** and passed to the web redirector **606**. At step **702**, the web redirector passes the message to an available authentication server **602**. As shown in step **703**, once an authentication server **602** validates a request, it passes the request through the main firewall **610** to a routing engine **110**. The routing engine **110** processes the request and returns a response to an authentication server **602** at step **704**. Routing engines **110** also accept detail reports from authentication servers **602**. Routing engines **110** forward transaction details, including the digitally-signed requests and detail reports to the database **120**, which may later be accessed by the billing and settlement system **124**. Most service points **112** use a virtual private network (VPN) link through the main firewall **610** for communication to the database **120**.

Once a routing engine **110** returns route information, the authentication server **602** adds authorization information to the response before returning it to the requesting device (gateway). Step **705**, in which the authentication server **602** responds to the requesting device, is the point at which authorization is added. When the routing engine **110** returns multiple eligible devices that can terminate the request, separate authorization information is created for each eligible device. This is true whether the devices are to be used simultaneously (such as in a multi-point conference) or

serially (in case the first choice is unavailable, for example). The originating device (source gateway **108**) must present the appropriate authorization information to a terminating device (destination gateway **114**) during call setup.

Authorization information consists of several pieces of information subjected to appropriate cryptographic transformations. The exact information depends on the particular service, but, in general, comprises: (1) sufficient information to uniquely identify the call, which may include the called and calling numbers, network addresses of the originating and terminating devices, unique identifiers such as call reference values and so on; (2) the transaction identifier, modified as necessary for terminating devices (for point-to-point services, for example, transaction IDs for terminating devices are changed from even to odd and their Hamming code is regenerated). Since terminating devices must include a transaction ID in detail reports, including a transaction ID in the authorization information forces the terminating device to examine that information and increases the likelihood that it will thoroughly check the information; (3) a valid time and an expiration time which limit the duration of call setup to help prevent inappropriate re-use of authorization information; and (4) a random value to be combined with the valid and expiration times for eliminating the probability of inappropriate reuse of authorization information. Terminating devices, upon accepting a call, are required to store this random number until the expiration time has passed. After the expiration time has passed, a terminating device must reject any setup request that includes the same random number. Authorization information may also include a maximum call duration, which limits the duration of calls that a device is willing to authorize. Authorization information may be encrypted using the public key of the terminating device and may be digitally signed by the service point **112**. The encryption prevents originating devices from modifying its contents, and the digital signature lets the terminating device verify that the information did come from the service point **112**.

The quality of the ultimate communication between originating and terminating devices is important. Round trip delay, for example, is a critical factor in the quality of voice phone calls. The service point operator **125** is able to estimate the communications quality to different terminating devices and use those estimates to rate each possible route. Some of the models used to estimate quality depend heavily on the specific service. Audio codecs, for example, may have a significant effect on voice quality. Some quality measures, however, apply generally to many services accessed over the Internet **102**. Those skilled in art will recognize that service quality monitoring may be accomplished by a separate system (not shown) connected to the Internet **102** and maintained by the service point operator **125**. Additionally, the mechanisms for performing service quality monitoring may be incorporated into a service point **112**, or any other system maintained by the service point operator **125**. Also, a source gateway **108** or a destination gateway **114** may independently perform the tasks of service quality monitoring. A service point operator **125** may provide service quality monitoring software to assist the source and/or destination gateway operator.

One measure of quality between two devices is the length of the network path between them. The length of the network path can be modeled by calculating the number of autonomous systems (AS) in the path between the two devices. To obtain the information needed to perform this calculation, software may be developed to establish Border Gateway Protocol (BGP4) peering relationships with BGP neighbors in other autonomous systems. From these peering sessions, a path may be determined, defined in terms of autonomous systems, from any peered AS to any other AS. To calculate exact paths, a service point operator **125** must peer with every transit AS that is a neighbor of an AS containing customer of the service point operator **125**. It is not necessary to peer with the customer's autonomous systems directly. In the absence of complete AS connectivity, the service point operator may estimate the distance between AS from partial information.

In some cases the difference in router hops between an originating device and multiple potential terminating devices may be estimated. To do so, a service point operator may use the "traceroute" command to calculate hop counts from a service point **112** or other central point to each device (gateway). Though this information does not provide an absolute hop count between the originating gateway **108** and terminating gateways **114**, the relative difference may be used to estimate the relative difference the originating gateway **108** would experience.

In a manner similar to relative hop counts, a service point operator **125** can also estimate relative differences in round trip delay. Instead of "traceroute," the service point operator may use UDP echo probes to measure round trip delays from a service point **112** or other central location. The difference in delays between two potential destination gateways **114** serves as an estimate of the delay difference an source gateway **108** would experience between the same two destination gateways **114**.

Another approach that a service point operator **125** may use to model expected service quality is extrapolation from historical data. Along with billing information, a service point operator **125** may collect quality measurements as part of detail reports. Traditional statistical techniques adapted for the Internet environment can then be used to project future service quality.

In an exemplary embodiment, the service point operator **125** may maintain a system for monitoring the service quality to all gateways participating in the service. The service point operator **125** may allow gateway operators to specify a premium they are willing to pay for improved service quality. The service point provider **125** may then program the service point **112** to route calls to gateways offering improved quality, as long as the rates charged fall within the originators' willingness to pay. If gateway operators are willing to make investments to improve their service quality, this approach lets them recover the cost of those investments through higher rates. Note that the service point operator **125** itself will likely not set rates or service quality standards. The concept of the service point **112** is to provide a market-based service that allows gateway operators the flexibility to set their own rates and establish their own service quality strictly according to market requirements.

The service point operator **125** may rely on two key parameters to determine Internet service quality: (1) delay and (2) packet loss. Although other factors can have a significant influence on users' perceptions of quality, these two parameters represent the most direct measures of the Internet's contribution to quality.

Round trip delay between the service point **112** and each gateway may be measured. The delay may be recorded in milliseconds with an accuracy of approximately 5 milliseconds. At each measurement, the paths used for the request and response messages are the normal forwarding paths employed by routers between the service point **112** and the gateways. Bi-directional packet loss between the service point **112** and each gateway may also be measured. Loss

may be recorded as a percentage, where, for example, a measurement of 10% indicates that 1 out of 10 packets transmitted did not reach its destination.

In collecting the data for delay and packet loss, the service point operator **125** simulates the environment of an H.323-compliant voice call as closely as possible. The collection techniques are also structured to maximize the statistical relevance of the measurements. To probe both delay and loss, the service point operator **125** may send echo messages to the gateways and measures the resulting responses. To more closely simulate voice traffic, UDP echo messages with 256 octets of user data may be used. These messages may be directed to UDP port 7 on the destination gateway. If no responses are returned, the service point operator **125** may assume that UDP echo service is not available on the gateway in question, and it may retry the measurement using ICMP Echo Requests (pings), also with 256 octets of user data.

Since ICMP messages are often subject to special processing in routers and hosts, ICMP measurements are likely to indicate greater delay than UDP messages. For this reason, the service point operator **125** may encourage all gateway operators to enable UDP echo service on their gateways. The service point operator **125** may choose to make no adjustments to measurements obtained via ICMP Echo exchanges, but rather may compare them directly to other measurements made through UDP.

For each measurement block, the service point operator generates a steady stream of UDP or ICMP traffic at an exemplary rate of 3 packets per second. This stream roughly simulates the talk interval of a G.723A codec commonly used for H.323-compliant gateways. Each block may be measured for bi-directional packet loss and average round-trip delay. Both to avoid self-synchronization effects, and to minimize statistical bias, delay nd loss measurements may be made according to a Poisson distribution. Both the interval between measurement blocks and the length of each block are random values that follow an exponential distribution. Adjustment may be made to the mean of these distributions to obtain the best balance between network loading and measurement reliability. Those skilled in the art will appreciate that the practice of conducting measurements at exponential intervals, is not meant to imply that voice traffic (or, indeed, any Internet traffic) may be modeled by a Poisson process. The choice of exponentially distributed intervals is intended purely to minimize time bias in the measurements, not to mimic traffic patterns.

To calculate a single number for delay and loss for each gateway, straightforward statistical processing may be used. In particular, measurements may be retained for one week, and a percentile ranking within the retained values may be relied upon. The retained data set may be updated (by discarding old data and adding new measurements) each day at 0:00 hours UTC. When a gateway first joins the service, no service quality measurements will be available until 0:00 UTC on its second day of operation. Until a full week has passed, the service point operator **125** will rely on the available data for quality determination. As a single, representative value for both delay and loss, the service point operator **125** may use the 80^{th} percentile for each measurement, over the most recent one week interval.

At the billing and settlement system **124**, a service point operator **125** provides net settlement and billing services for its customers. The billing and settlement system **124** may be located at central and secured facilities. Service points **112** periodically update the billing and settlement system **124** with detail reports received from devices (gateways). The billing and settlement system reconciles the different reports from each device involved in a single communications transaction and calculates the net settlement funds to be paid to or collected from each customer. The service point operator **125** may execute the actual monetary transactions though traditional financial networks **126** at various financial institutions **128**.

Exemplary Routing Engine

As described above, a service point **112** in the exemplary operating environment includes a routing engine **110** that is responsible for providing routing information to a source gateway **108**. The exemplary routing engine **110** is particularly useful in situations where a source gateway **108** is provided with the identity of the called party **118** (i.e. the called telephone number), but is not provided with the address of an appropriate destination gateway **114**. Use of the exemplary routing engine **110** is even more appropriate in situations where there are more than one eligible destinations gateway **114***a–c*. The exemplary routing engine **110** has the ability to determine the network addresses of all eligible destination gateways **114***a–c* and then prioritize the eligible destination gateways **114***a–c*. As mentioned, the exemplary routing engine **110** provides the prioritized list of destination gateways **114***a–c* to the source gateway **108**. The source gateway **108** then attempts to setup a call with the first-listed destination gateway **114**. If the first-listed destination gateway **114** does not accept the call, for any reason, the originating gateway **108** attempts to setup a call with the next-listed destination gateway **114**. The source gateway continues to attempt to setup the call with each successive next-listed destination gateway **114** until the call is established.

Prioritization of eligible destination gateways **114** by the exemplary routing engine **110** is based on (1) preferences established by a source gateway operator **109**, (2) monetary value charged by a destination gateway operator **115** for access to a destination gateway **114** and (3) network environment conditions. The exemplary routing engine **110** provides maximum flexibility in routing choices by allowing source gateways operators **109** to set their own preferences and destination gateways operators **115** to set their own costs. Network environment conditions may be evaluated for the routing engine by a separate system, as described above in connection with service quality monitoring. Preferences, costs and network environment conditions are stored in a database **120** and this information is periodically accessed by the routing engine **110**.

A source gateway operator **109** may upload information to the database **120** via a web-site **122**, or via any other electronic transfer means. The web site **122** may be accessed via any commonly known web browser. The web site **122** may contain a form to be populated by the source gateway operator **109**. Those skilled in the art will recognize that no particular user interface (UI) for the web site **122** is required, however, the UI should be as user friendly as possible.

The preferences defined by the source gateway operator **109** relate to monetary cost, delay and reliability. The routing engine **110** uses the preferences set by the source gateway operator **109** as filters for eliminating potential destination gateways and determining determine the best destination gateway to terminate a called number. The source gateway operator **109** may specify none or any combination of preferences as its filters. Also, service point operator **125** and source gateway operator **109** may specify the maximum number of call routes that will be returned in response to each call authorization request. The routing engine **110** may be programmed to respond with whichever maximum number is less.

In the exemplary embodiment, a first preference is defined as "the maximum amount the source gateway operator **109** is willing to pay for a call to a specific dial string." All destination gateways charging rates that are greater than the maximum price are eliminated from the search for the optimal call route. The maximum price criteria may be specified as a function of time of day and day of the week. This maximum amount may be expressed in any type of currency and any fraction thereof.

Another preference in the exemplary embodiment may be defined as "the maximum delay a source gateway operator **109** is willing to tolerate." The maximum delay is preferably the overall network delay, which is measured by the time taken for a signal to travel between the calling party **104** and the called party **118**. The lower the network delay from when the calling party **104** speaks to when the called party **118** hears the words, the higher the quality of the conversation. Those skilled in the art will appreciate that there are many other factors that determine delay or latency in a voice telephone call. Other examples of delay include: delay due to interlocking of a digital conversation, buffering delays inside gateways, delays on public switched telephone networks (PSTN), etc. It is contemplated that the such other sources of delay may be factored into the "maximum delay preference." However, it is expected that network delay will be the most significant contributor to the overall quality of an Internet telephony call. Thus, in the exemplary embodiment, other sources of delay are ignored.

Another preference defined in the exemplary embodiment is the "maximum autonomous system (AS) hop count that the source gateway operator will tolerate." The Internet **102** comprises a collection of "autonomous" IP networks. Thus, a voice signal traveling from a source gateway **108** to a destination gateway **114** may traverse one or more autonomous systems. The fewer autonomous systems that a signal must traverse, the lower the network delay should be. While it is not necessarily true that a lower AS hop count will lead to lower delay, AS hop count does provide a good estimation of network delay. Furthermore, a lower AS hop count tends to suggest that there will be less signal loss (packet loss) when the voice signal reaches its destination.

A determination of AS hop count is instantaneous and may be derived from information relating to the dynamic topology of the Internet **102**, which is dictated by congestion, etc., that is continuously gathered and stored in the database **120**. To the contrary, network delay may only be determined by actual measurement, as described above, which involves significantly more time than an AS hop count calculation. Therefore, a source gateway operator **109** may elect to use the AS hop count preference, rather than the "maximum delay" preference.

An additional preference may be defined as "autonomous system (AS) matching," which dictates that, whenever possible, a route should be chosen such that both the source gateway **108** and the destination gateway **114** are on the same AS. A determination of AS matching is similar to a determination of AS hop count. A determination of AS matching dictates that given the choice of an AS hop count of zero and any other AS hop count, the route having the AS hop count of zero will be chosen. Similarly, "domain matching" and "platform matching" preferences may be defined, such that no destination gateway **114** that operates in a specified domain or on a specified platform will be selected to terminate a call.

Of course, a source gateway operator **109** may set as many or as few preferences as it would like. It is contemplated that preferences other than the exemplary preferences described herein may be implemented. For example, the source gateway operator **109** may also set preferences defining that all destination gateways **114** that are not interoperable with the source gateway **108** or do not offer the requested type of service, i.e. voice or fax, are to be eliminated from consideration. Other preferences may include, but are not limited to: "historical availability," which eliminates from consideration all destination gateways **114** that have historical availability less than the required availability specified by the source gateway operator **109**; "preferred operator," which eliminates from consideration all destination gateways **114** that are not operated by a preferred operator specified by the source gateway operator **109**; "packet loss," which eliminates from consideration all destination gateways **114** whose historical packet loss is greater than the minimum specified by the source gateway operator **109**; "latency," which eliminates from consideration all destination gateways **114** whose historical packet loss is greater than the maximum latency specified by the source gateway operator **109**; "quality of service (QoS) score," which eliminates from consideration all destination gateways **114** whose QoS is less than the minimum specified by the source gateway operator **109**; "RSVP preference," which eliminates from consideration all destination gateways **114** that cannot support, or are on networks that do not support, bandwidth reservation; and "best worst case," which eliminates from consideration all destination gateways **114** whose best worst case scenario for packet loss and latency exceeds the minimum best worst case scenario specified by the source gateway operator **109**. The best worst case is estimated by summing the packet loss or latency between the source gateway **109** and a reference point maintained by the service point operator **125** (SP_{ref}) plus the latency and packet loss between the destination gateway **114** and SP_{ref}. For example, the worst case for packet latency between a source gateway **109** and a destination gateway **114** is assumed to be equal to packet latency between the source gateway **109** and SP_{ref}+ destination gateway and SP_{ref}.

Preferences may also be ranked by the source gateway operator **109**, such that one type of preference is given more weight by the routing engine **110** when eligible destination gateways are prioritized. A predetermined system for ranking preferences is useful when the routing engine **110** locates more than one destination gateway **114** that satisfies all preferences. Thus, a ranking system may be used as a "tie breaker." A source gateway operator **109** may prioritize its preferences in any order, or no order at all.

Preferences may be ranked by least cost. By way of example, a preference designated by a source gateway operator **109** may dictate that the maximum price the source gateway operator **109** is willing to pay for a call to London is $0.40/minute. The routing engine **110** may locate two eligible destination gateways **114***a–b* to terminate the call; one destination gateway **114***a* charging $0.35/minute and the other destination gateway **114***b* charging $0.30/minute. Both destination gateways **114***a–b* are eligible because they each meet the preference designated by the source gateway operator **109**. However, the source gateway operator **109** may also specify that all eligible destination gateways **114***a–b* are to be ranked (or sorted) by least cost. Thus, the destination gateway **114***b* charging $0.30/minute will be assigned a higher priority than the destination gateway **114***a* charging $0.35/minute.

Preferences may also be ranked by AS matching, such that priority is given to routes involving gateways on the same autonomous system. Also, preferences may be ranked by

subscriber (intra-domain) matching, such that priority is given to routes involving destination gateways **114** connected to the same network as the source gateway **109**. Intra-domain routing may be predefined to take first priority even if price and quality of service are inferior to other gateways.

Preferred platform matching allows a source gateway operator **109** to prioritize its preferred destination gateway platform for termination. For example, Lucent and VocalTec gateways may be interoperable; however, a source gateway operator **109** who has deployed Lucent gateways might prioritize that calls be routed to Lucent gateways as a first choice. Or the source gateway operator **109** might specify that VocalTec gateways be eliminated as a possible destination gateways, even though they are compatible and are the best match for the source gateway operator's **109** other routing criteria.

Preferences may also be ranked based on minimum AS hop count. Priority may be given to BGP query route calls to destination gateways **114** that can be reached with the fewest autonomous system hops. Preferences may also be ranked based on historical records of availability (as a function of day of week and time of day). Priority may thus be given to destination gateways **114** with the best historical availability or eliminate those gateways with historical availability less than the minimum required by the source gateway operator **109**. Further, to support. implementation of bilateral agreements, a service point operator **125** may allow source gateway operators **109** to prioritize their preferred destination gateway operators **115** for termination. For example, source gateway **109** may specify that destination gateway operator **115***b* is always its first choice for termination and destination gateway operator **115***c* is its second choice for termination even if other terminating gateways are a better fit for the source gateways operator's **109** routing criteria. Similarly, a source gateway operator **109** may prioritize specific individual destination gateways as its preferred termination point for a call to a dialed number string.

Based on historical records of packet loss between the source gateway **109** and potential destination gateways **114** (possibly as a function of day of week and time of day), priority may be assigned based on lowest historical packet loss. In a similar manner, priority of routing may be assigned based on the lowest historical packet latency between the source gateway **109** and potential destination gateways **114** (possibly as a function of day of week and time of day). Further, ranking of potential destination gateways **114** may be based on other preference, such as: QoS scoring, by using historical data of packet loss, latency and availability with codec and gateway implementation choices to make best destination gateway **114** selection; RSVP, by routing calls based on which call path offers the required bandwidth reservation at the lowest price; and "best worst case," by routing calls based on the best worst case scenario described above. In the absence of any ranking or sorting scheme specified by a source gateway operator **109**, the routing engine **110** may either employ its own ranking scheme, or in the interest of impartiality, randomly prioritize the eligible destination gateways **114**.

When setting preferences, the source gateway operator **109** may also provide the routing engine **110** with "preference criteria," which define the circumstances in which a given preference or set of preferences are to apply. For example, the source gateway operator **109** may specify a called number prefix to which a certain preference is to apply. Accordingly, all calls placed to the specified called number prefix are to be routed based upon the corresponding preferences. In the exemplary embodiment, a called number prefix is defined by ITU standard E.164. An exemplary called number may be "+1 404 567 8910." According to E.164, the components of the exemplary called number may be defined as follows: "+" designates an international call; "1" is the country code for the United States; "404" is the national destination code, also known as the area code in the United States; "567" is the local exchange; and "8910" is the identifier of the actual telephone line.

The source gateway operator **109** may designate any portion of the called number as the called number prefix. By way of illustration, the called number prefix "+1" signifies that a certain preference is to be applied to all calls to/from the United States. A called number prefix of "+1404" signifies that a certain preference s to be applied to all calls to/from Atlanta. Similarly, if it is known that corporation X utilizes all telephone numbers between +1.404.567.8910 and +1.404.567.8919, the called number prefix may be designated as +1404567891, which will cause a certain preference to be applied to all calls to/from corporation X. As can be seen, the called number prefix may comprise any portion of the called number and is not limited to any particular division thereof.

In addition to called number prefixes, a preference criteria may include the identity of a specific source gateway **108**. As mentioned previously, a source gateway operator **109** may own and/or operator many source gateways **108**. The source gateway operator **109** may desire that different preferences apply to calls handled by different source gateways **108**. Other preference criteria may include specifications of the time of day and/or the day of the week. A time of day preference criteria may be specified down to the second (or any fraction thereof), if desired. An effective date may also be specified as a preference criteria. An effective date allows a source gateway operator **109** to specify that certain preferences are to be considered for routing from a given point in time.

Routing decisions made by exemplary routing engine **110** are also based on preferences set by destination gateway operators **115**. In the exemplary embodiment, destination gateways operators **115** set only one preference. The preference set by a destination gateway operator **115** is the monetary charge for access to a destination gateway **114**. A destination gateway operator **115** is typically not concerned with delay, packet loss, or other factors that may be of concern to source gateway operator **109**. Once a network packet containing a voice signal arrives at a destination gateway **114**, the packet is routed to the appropriate called party **118**, regardless of the amount of delay or packet loss associated therewith. Like the originating gateway operator **109**, the destination gateway operator **115** may designate certain preference criteria. Such preference criteria allow the destination gateway operator **115** to define the circumstances in which its preferences, i.e., cost schedule, will apply. Preference criteria may relate to a specific destination gateway **114**, a called number prefix, a time of day, a day of the week, an effective date, etc.

All preferences and preference criteria set by the source gateway operator **109** and the destination gateway operators **114** are stored in the database **120**. FIG. **8** illustrates an exemplary database table **801** for storing information relating to a source gateway **108**. In an exemplary embodiment, database tables **801** may be sorted first by gateway identification number **802** and then by called number prefix **804**, effective date **806** (in descending order), start day **808** and start hour **810**. Sorting data in the described manner is not

required, but is recommended for improved performance. A routing engine **110** may be required to process a significant amount of routing requests in a short period of time. Thus, the database table **801** should be organized in an easily searchable manner. The ultimate goal of the routing engine **110** is to access the database table **801** to determine the preferences for a source gateway **108**, such as the maximum price **820** that the source gateway **108** is willing to pay for an Internet telephony call.

FIGS. **9**A and **9**B describe an exemplary method by which a routing engine **110** may access a database table **801** to locate the preferences for a source gateway **108**. At step **905**, a outing engine **110** receives an authorization request from a source gateway **108**, via an authentication server **602**. The authorization request includes the identification number **802** of the source gateway **108**. At step **910**, the routing engine **110** accesses the database table **801** to locate the preferences corresponding to the source gateway identification number **802**. First at step **915**, a search of the appropriate database column is conducted to locate the source gateway identification number **802**. A binary search is a quick and effective method of searching for the desired gateway identification number **802**.

At step **920**, an offset value **826** and a 'number of entries' value **824** are checked to determine the location of the first and last set of preferences for a given source gateway identification number **802**. Many times, a source gateway operator **109** will designate multiple sets of preferences, each to be applied in circumstances defined by designated preference criteria. Therefore the database table **801** maintains an offset value **826** so that the routing engine is able to locate the first and last record for a given gateway identification number **802**.

Once the first and last record for the desired gateway identification number **802** is located, a search is conducted at step **925** on all included entries to determine the longest called number prefix **804** corresponding to the gateway identification number **802**. The size **816** of the longest called number prefix may be stored in the database table **801** for searching convenience. Again, a binary search is likely to be the most efficient way to search for the longest called number prefix. Once the longest called number prefix **804** is located, it is checked at step **930** to see if it matches the called number for the Internet telephony call, as supplied by the source gateway **108**. If, at step **930**, all digits in the longest called number prefix **804** are not found in the called number the called number prefix is not considered to match the called number. In that case, the preferences corresponding to the longest called number prefix **804** will not apply to the call and the next longest called number prefix **804** is located at step **940**. The matching process is repeated until a matching called number prefix **804** is located.

When a matching called number prefix **804** is located, a determination is made at step **945** as to whether the source gateway **108** has a rate plan corresponding to the called number prefix. If no rate plan is found, the method returns to step **940** to locate the next-longest called number prefix **804**. If a rate plan is found, however, a determination at step **950** is made as to whether the called number is excluded from the rate plan. If the called number is excluded from the rate plan, the method again returns to step **940** to locate the next-longest called number prefix **804**.

If it is determined that the called number is not excluded from the rate plan, a determination is made at step **955** as to whether the source gateway **108** has received permission from the service point operator **125** to access the service point. Permission may be based on a finding that the source gateway operator **109** has secured sufficient funds. If permission has not been received by the source gateway **108**, the method exits at step **975**. Proceeding, a determination is next made at step **960** as to whether the rate plan of the source gateway **108** is effective. If the effective date of the rate plan is in the future, the preferences set by the source gateway **108** cannot yet be applied. Similarly, at step **965**, the start day, **808** start hour **810**, end day **812** and end hour **814** are checked to determine validity with respect to the current day and time. If the rate plan is not in effect or the time of day or day of week are not valid, the method returns to step **940** to locate the next-longest called number prefix. When a record is encountered that satisfies all called number prefix **804** and date and time requirements, the corresponding price preference **820** is considered to apply to the call. This price preference is read at step **970** and the method terminates at step **980**. Those skilled in the art will recognize that the above described method may be used to locate other preferences of the source gateway **108**, such as a delay preference, an AS hop count preference and, but not limited to, any other preference mentioned above.

Once the preferences of the source gateway **108** are located, the routing engine **110** must locate an eligible destination gateway **114**. FIG. **10** shows that a database table **1001** for storing information relating to a destination gateway **114** may be sorted by terminating number prefix **1002** and then by destination gateway identification number **1004**. In this way, the database table **1001** may be conveniently searched by the "trie" method. It is anticipated that the database table **1001** will store up to several millions of entries. It has been determined that a trie method is the most efficient way to access the appropriate data. However, as will be appreciated by those of ordinary skill in the art, any data access method may be employed.

FIGS. **11** A, **11**B and **11**C show an exemplary method that may be used by the routing engine **110** to locate eligible gateways in a database table **1001**. First, at step **1102**, the routing engine **110** accepts the called number from the source gateway **108**. Next, at step **1104**, the routing engine **110** searches for terminating number prefixes **1002** that match the called number, so as to locate a plurality of destination gateways **114** that may be able to terminate the call. As mentioned, an effective method for locating matching terminating number prefixes **1002** in the database is a well-known trie function.

If no matching terminating number prefixes **1002** are located at step **1105**, the method ends at step **1150**. For each matching terminating number prefix **1002** that is located, the corresponding destination gateway may be put through an set of initial "pre-screening" tests in order to determine if it is a potential destination gateway that is able to terminate the call. At step **1106**, a destination gateway is identified that corresponds to a matching terminating number prefix **1002**. Then, at step **1107** a determination is made as to whether the destination gateway **114** is functionally able to terminate the call. For one reason or another, a given destination gateway **114** may be disconnected, disabled, or otherwise taken off-line. Such off-line destination gateways are eliminated from consideration as potential destination gateways **114**. Thus, if the destination gateway **114** is not functionally able to terminate the call, the method proceeds to step **1224**, where a determination is made as to whether any other matching terminating number prefixes **1002** were found.

At step **1108** a determination is made as to whether the terminating gateway **114** has received permission from the service point operator **125** to conduct transactions with the routing engine. To reiterate, permission may be based on any

subjective factors, including the amount of funding available to the destination gateway **114**. If permission has not been granted by the service point operator **125** the method proceeds to step **1124**, where a determination is made as to whether any other matching terminating number prefixes **1002** were found.

If the destination gateway **114** has been granted permission, a determination is made at step **1110** as to whether the destination gateway **114** is interoperable (compatible) with the source gateway **108**. If the destination gateway **114** is not interoperable with the source gateway **108**, the method proceeds to step **1124**, where a determination is made as to whether any other matching terminating number prefixes **1002** were found.

If the destination gateway **114** is compatible with the source gateway **108**, a determination is made at step **1112** as to whether the destination gateway **114** has a rate plan in effect. Again, if no effective rate plan exists, a search is made for additional matching terminating number prefixes **1002** at step **1124**. If an effective rate plan exists, a determination is made at step **1114** as to whether the requested type of service is supported. As mentioned, the source gateway **108** may request various services such as voice, fax, data, etc. If the destination gateway **114** does not support the requested service, the method proceeds to step **1124** to search for another potential destination gateway **114**.

However, if the destination gateway **114** does support the requested service, a determination is made at step **1118** as to whether the rate plan specifies a termination price for the present time and date. Once again, if the termination price is not valid for the present time and date, the method proceeds to step **1124** to search for another matching terminating number prefix. Otherwise, a determination is made at step **1120** as to whether the called number is excluded or "blocked" by the destination gateway **14**. As an example, a destination gateway may be programmed to terminate all calls to the United States, except "1-976" calls. If the called number is excluded, the method moves to step **1124** to search for another matching terminating number prefix. If the called number is not excluded, the destination gateway **114** has passed all of the "pre-screening" tests and is considered to be a potential destination gateway. The terminating number, address and other information relating to the potential destination gateway **114** is stored in a "first-pass" list at step **1122**.

Next at step **1124**, a determination is made as to whether any other matching terminating number prefixes **1002** were found. If so, the above described steps are repeated until all potential destination gateways **114** have been added to the first-pass list. Otherwise, the method proceeds to step **1130** where the first-pass list is sorted by destination gateway identification number **1004** and by the number of digits in the terminating number prefix **1002** (in descending order). Then, at step **1132**, the entries for the first-listed potential destination gateway are searched to locate the longest terminating number prefix **1002** for that destination gateway that matches the called number.

At step **1134**, a determination is made as to whether the potential destination gateway **114** is able to terminate the call to the longest matching terminating number prefix while satisfying the preferences set for the source gateway **108**. If the potential destination gateway is able to terminate the call while satisfying the source gateway **108** preferences, the potential destination gateway **114** information is stored in a second-pass list of remaining potential destination gateways **114** at step **1138**. Subsequently, at step **1140**, a determination is made as to whether any other potential destination gateways are in the first-pass list. If so, the entries for the next potential destination gateway are searched to locate the longest terminating number prefix. The method then returns to step **1134** and is repeated for every potential destination gateway in the first-pass list.

When complete, the second-pass list of remaining potential destination gateways **114** is sorted at step **1142** according to a pre-determined ranking system. As mentioned, the ranking system may be designated by the source gateway operator **109** and may be such that a certain weight is to be accorded to each source gateway preference. Alternately, the ranking system may be designated by the service point operator **125**. At step **1144** all potential destination gateways **114** having a duplicate rank are sorted in random order. Lastly, at step **1148**, a configurable number of the prioritized potential destination gateways **114** are returned to the source gateway **108**. The number of returned prioritized potential destination gateways may be specified by the source gateway operator **109** or by the service point operator **125**. The exemplary method terminates at step **1150**.

As shown in FIG. **12**, the internal architecture of an exemplary routing engine **110** comprises several interrelated components. These components will be described in detail to provide a further understanding of how to implement the exemplary routing engine **110**. However, those skilled in the art should recognize that the architecture of FIG. **12** is provided by way of example only and is not intended to limit the scope of the invention to components shown.

A call router **1202** is the component that is actually responsible for receiving call authorization requests and returning a set of eligible terminating gateways to the authentication server. It is possible to run several call routers **1202** at the same time on a single routing engine **110**. A call router **1202** can be configured to use one or more threads to process incoming transactions. Also, the call router **110** can be configured not to use threads at all, effectively acting as a single-threaded program.

In order to determine how to route a call, a call router **1202** needs information regarding the source gateway **108** and the destination gateway **114** such as pricing, exchange rates and other preferences. This information is stored in a central database **120** and is periodically transferred to a local database **1204** that resides on every routing engine **110**. For performance reasons, the call router **1202** does not access the local database **1204** directly. Instead, the call router **1202** uses a set of reference files **1208** that contain all information needed to route calls, but in a format that is much faster to access than a database table. The reference files are created periodically by the reference file extractor **1210**. The reference file extractor is also responsible for notifying all call routers **1202** when a new set of reference files **1208** is created. In addition, all call routers **1202** periodically (a few times per hour) check for the existence of a new set of reference files **1208**, just in case a message from the reference file extractor **1210** is lost. There is no requirement for all call routers **1202** to switch to a new set of reference files **1208** at exactly the same time, due to the fact that all information contains effective dates, which must be at least twenty-four hours in the future.

Each call router **1202** memory-maps the reference files **1208** in read-only, shared mode. This means that all processes and threads using the same set of reference files **1208** will effectively share memory. Thus, if a total of ten call routers **1202** are accessing a set of reference files **1208** occupying 60 MB, the total memory usage will be 60 MB and not 10*60 MB.

As mentioned, the reference file extractor **1210** is responsible for creating reference files **1208** used by the call router

1202 to decide how to route calls. The reference file extractor 1210 extracts information from the local database 1204 and creates a set of four files: a first file relating to originating gateways, their pricing criteria, numbers, etc.; a second and third file relating to terminating gateways, their pricing criteria, numbers, etc.; and a fourth file relating to exchange rates. Either the whole set of reference files 1208 is successfully created or not at all. The base file name is the same for every file in a set, the file extensions are different. A unique base file name is generated for every set. Once a set of reference files 1208 is created, the file extractor 1210 notifies all call routers 1202 that new files are present.

All references between data entities within each reference file 1208 as well as across files are expressed as record numbers relative the beginning of the file being referenced. That means the same set of reference file 1208 can be used at the same time by different processed in read-only mode.

In one implementation, the file extractor 1210 may completely re-create all reference files 1208 every time it runs; (e.g. once/hour), provided that data in the tables has changed since the last execution. It take at most thirty seconds (on Ultra2) to completely re-create all information for 5000 destination and source called numbers (total 10000 numbers); that includes the time required to retrieve the data from the local database 1204. All sorting, etc. is done by the file extractor 1210 itself, not by the local database 1204.

Another implementation of the file extractor may create incremental reference file sets 1208 to be merged with previous sets, so as to create new versions. Alternately, the file extractor 1210 may construct an index for accessing every reference file 1208 or store the data as linked lists to allow for very frequent insertions and deletions without having to physically sort any data.

The file extractor 1210 is also responsible for near-real time deactivation of gateways. When instructed to do so, the file extractor 1210 may update relevant reference files 1208 in each set to indicate that a gateway has been deactivated. The file extractor 1210 updates reference files 1208 by memory mapping the reference files 1208 in shared, writeable mode, looking up entries belonging to the gateway to be deactivated and modifying the entries. When finished, the file extractor makes the changes visible to other programs, e.g. to the call router 1202. The process of deactivating gateways may be executed concurrently with the execution of call routers 1202 and does not require any explicit synchronization between such components.

While this invention has been described in detail with particular reference to preferred embodiments thereof, it will be understood that variations and modifications can be effected within the spirit and scope of the invention as described hereinabove and as described in the appended claims.

We claim:

1. A method for determining a preferred route for a call that is to be routed over an IP network comprising:

accepting from a source gateway operator a preference corresponding to a source gateway;

accepting a called number from the source gateway;

identifying a plurality of potential destination gateways that are capable of terminating the called number;

filtering the potential destination gateways based on the preference from the source gateway, so as to yield one or more remaining destination gateways;

prioritizing the remaining destination gateways according to a predetermined ranking system;

supplying the addresses of the prioritized remaining destination gateways to the source gateway.

2. The method of claim 1, wherein the source gateway successively attempts to establish the call with each prioritized remaining destination gateway until the call is established.

3. The method of claim 1, wherein the source gateway operator also sets a preference criteria that indicates circumstances in which the preference is to be applied.

4. The method of claim 1, wherein the preference relates to a determination of autonomous system matching.

5. The method of claim 1, wherein the preference relates to a determination of domain matching.

6. The method of claim 1, wherein the preference relates to a determination of specified platform system matching, the platform being specified by the source gateway operator.

7. The method of claim 1, wherein the preference relates to a maximum autonomous system hop count.

8. The method of claim 1, further comprising the step of randomizing any of the prioritized remaining destination gateways that have a duplicate priority.

9. The method of claim 1, wherein the source gateway operator designates a number of the addresses of prioritized remaining destination gateways to be supplied to the source gateway.

10. The method of claim 1, wherein the call comprises a voice over IP call.

11. The method of claim 1, wherein the call comprises a video over IP call.

12. The method of claim 1, wherein the call comprises a fax over IP call.

13. A routing engine for assisting a source gateway in determining a preferred route for a call that is to be routed over an IP network comprising:

a database for storing a set of source gateway preferences and a set of destination gateway preferences; and

a call router operable to:

accept a called number from the source gateway,

identify a plurality of potential destination gateways that are capable of terminating the call to the called number,

filter the potential destination gateways based on the source gateway preferences and the destination gateway preferences,

prioritize remain destination gateways based on a predetermined ranking system set by a source gateway operator, and

supply addresses of the prioritized remaining destination gateways to the source gateway.

14. The routing engine of claim 13 comprising a plurality of call routers.

15. The routing engine of claim 13, wherein the source gateway and a destination gateway are operated independently of each other.

16. The routing engine of claim 13, wherein the set of source gateway preferences are supplied by a source gateway operator via a web-site and are downloaded into the database.

17. The routing engine of claim 13, wherein the set of destination gateway preferences are supplied by a destination gateway operator via a web-site and are downloaded into the database.

18. The routing engine of claim 13, wherein the source gateway preferences relate to a maximum cost that the source gateway is willing to pay for the call.

19. The routing engine of claim 13, wherein the destination gateway preferences relate to a rate that the destination gateway will charge for the service of terminating the call.

20. The routing engine of claim 13, wherein the call comprises a voice over IP call.

21. The routing engine of claim 13, wherein the call comprises a video over IP call.

22. The routing engine of claim 13, wherein the call comprises a fax over IP call.

23. A method for routing a call over an IP network comprising:

supplying a source gateway preference and a called number to a third-party routing engine, the routing engine operable to:

locate a plurality of potential destination gateways for terminating the call,

filter the potential destination gateways based on the source gateway preference, so as to yield a set of remaining potential destination gateways, and

prioritize the remaining potential destination gateways;

receiving from the routing engine a list of the prioritized remaining potential destination gateways; and

in order of descending priority, successively attempting to route the call to each of the listed remaining potential destination gateways until the call is established.

24. The method of claim 23, wherein the source gateway preference relates to a maximum cost that the source gateway is willing to pay for the call.

25. The method of claim 23, wherein the source gateway preference relates to a determination of autonomous system matching.

26. The method of claim 23, wherein the source gateway preference relates to a determination of domain matching.

27. The method of claim 23, wherein the source gateway preference relates to a determination of specified platform matching.

28. The method of claim 23, wherein the source gateway preference relates to a maximum number of autonomous system hops that the source gateway will tolerate for the call.

29. The method of claim 23, wherein the source gateway preference relates to a determination of historical availability of the potential destination gateways.

30. The method of claim 23, wherein the source gateway preference relates to a preferred destination gateway operator.

31. The method of claim 23, wherein the source gateway preference relates to a preferred destination gateway.

32. The method of claim 23, wherein the source gateway preference relates to a maximum amount of packet loss that will be tolerated by the source gateway during the call.

33. The method of claim 23, wherein the source gateway preference relates to a maximum amount of latency that will be tolerated by the source gateway during the call.

34. The method of claim 23, wherein the call comprises a voice over IP call.

35. The method of claim 23 wherein the call comprises a video over IP call.

36. The method of claim 23, wherein the call comprises a fax over IP call.

37. A method for determining a preferred route for a call that is to be routed over an IP network comprising:

accepting from a source gateway operator a preference corresponding to a source gateway, wherein the preference relates to a maximum price that a source gateway is willing to pay for the call;

accepting a called number from the source gateway;

identifying a plurality of potential destination gateways that are capable of terminating the called number, wherein the plurality of potential destination gateways each supply a rate plan;

filtering the potential destination gateways based on the preference from the source gateway, so as to yield one or more remaining destination gateways, wherein filtering the potential destination gateways comprises eliminating all potential destination gateways charging a rate that is greater than a maximum price that the source gateway is willing to pay;

prioritizing remaining destination gateways according to a predetermined ranking system;

supplying the addresses of the prioritized remaining destination gateways to the source gateway.

38. The method of claim 37, wherein prioritized a plurality of remaining destination gateways comprises ranking the plurality of remaining destination gateways from the lowest rate charged for terminating the call to the highest rate charged for terminating the call.

39. The method of claim 38, wherein a plurality of remaining destination gateways are further ranked according to autonomous system matching.

40. A method for determining a preferred route for a call that is to be routed over an IP network comprising:

accepting from a source gateway operator a plurality of preferences and a plurality of preference criteria corresponding to a source gateway;

accepting a called number from the source gateway;

identifying a plurality of potential destination gateways that are capable of terminating the called number;

filtering the potential destination gateways based on the preference from the source gateway, so as to yield one or more remaining destination gateways:

prioritizing the remaining destination gateways according to a predetermined ranking system;

supplying the addresses of the prioritized remaining destination gateways to the source gateway; and

determining from the preference criteria as to which of the preferences apply to the source gateway initiating the call.

41. The method of claim 40, wherein identifying the plurality of potential destination gateways comprises identifying a plurality of destination gateways that are able to terminate any portion of the called number.

42. A method for determining a preferred route for a call that is to be routed over an IP network comprising:

accepting from a source gateway operator a preference corresponding to a source gateway;

accepting a called number from the source gateway;

identifying a plurality of potential destination gateways that are capable of terminating any portion of the called number;

filtering the potential destination gateways based on the preference from the source gateway, so as to yield one or more remaining destination gateways;

prioritizing the remaining destination gateways according to a predetermined ranking system;

supplying the addresses of the prioritized remaining destination gateways to the source gateway.

43. The method of claim 42, further comprising the step of eliminating from consideration each of the destination gateways that do not have a rate plan in effect for the time and date of the call.

44. The method of claim 42, further comprising the step of eliminating from consideration each of the destination gateways that do not support the service requested by the source gateway.

45. The method of claim 42, further comprising the step of eliminating from consideration each of the destination gateways that do not operate with the source gateway.

46. The method of claim 42, further comprising the step of eliminating from consideration each of the destination gateways that have not been granted permission to terminate the call.

47. A method for routing a call over an IP network comprising:

supplying a source gateway preference, a called number, and a pre-determined ranking system to a third-party routing engine, the routing engine operable to:

locate a plurality of potential destination gateways for terminating the call,

filter the potential destination gateways based on the source gateway preference, so as to yield a set of remaining potential destination gateways, and

prioritize the remaining potential destination gateways according to the ranking system;

receiving from the routing engine a list of the prioritized remaining potential destination gateways; and

in order of descending priority, successively attempting to route the call to each of the listed remaining potential destination gateways until the call is established.

48. The method of claim 47, wherein a plurality of source gateway preferences are supplied to the routing engine; and wherein the ranking system specifies a weight that is to be accorded to each source gateway preference when prioritizing th remaining potential designation gateways.

49. The method of claim 48, wherein the ranking system specifies that a highest priority is to be assigned to a least cost source gateway preference.

* * * * *

US006006269A

United States Patent [19]

Phaal

[11] **Patent Number: 6,006,269**

[45] **Date of Patent: Dec. 21, 1999**

[54] **ADMISSION CONTROL SYSTEM WITH MESSAGES ADMITTED OR DEFERRED FOR RE-SUBMISSION AT A LATER TIME ON A PRIORITY BASIS**

[75] Inventor: **Peter Phaal**, San Francisco, Calif.

[73] Assignee: **Hewlett-Packard Company**, Palo Alto, Calif.

[21] Appl. No.: **09/038,868**

[22] Filed: **Mar. 11, 1998**

[51] **Int. Cl.**[6] **G06F 13/00**

[52] **U.S. Cl.** **709/227**; 709/103

[58] **Field of Search** 709/227, 228, 709/229, 217, 219, 226, 104, 105, 103, 225, 235, 232, 206, 207; 710/244, 241, 240, 39, 36, 38, 40; 711/9, 10, 2, 1

[56] **References Cited**

U.S. PATENT DOCUMENTS

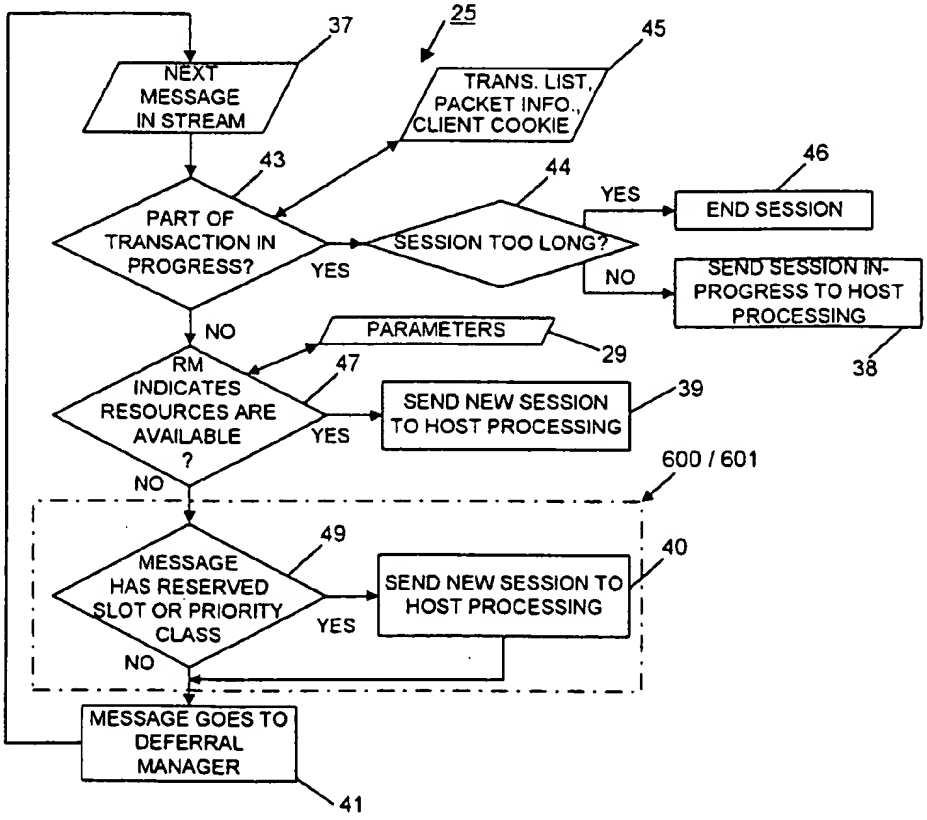
5,006,978	4/1991	Neches	364/200
5,218,676	6/1993	Ben-Ayed et al.	395/200
5,377,354	12/1994	Scannell et al.	395/650
5,481,312	1/1996	Cash et al.	348/466
5,617,541	4/1997	Albanese et al.	395/200.13
5,623,603	4/1997	Jiang et al.	395/200.04
5,699,521	12/1997	Iizuka et al.	395/200.15
5,742,588	4/1998	Thornberg et al.	370/236
5,754,752	5/1998	Sheh et al.	395/182.02
5,799,002	8/1998	Krishnan	370/234
5,826,031	10/1998	Nielsen	395/200.63
5,881,238	3/1999	Aman et al.	395/200.56
5,884,033	3/1999	Duvall et al.	395/200.36
5,889,951	3/1999	Lombardi	395/200.49

Primary Examiner—Mehmet B. Geckil
Attorney, Agent, or Firm—Marc P. Schuyler

[57] **ABSTRACT**

This disclosure provides for admission control having enhanced quality of service. A server-resident admission control system implements measurement-based admission control to determine whether a requested web site is available to process a new session. If the site is not available, based upon current resources and defined load parameters, the server-based system determines when the associated server can later provide preferred access to the client, and transmits to the client an indication of that time, together with a key. One example uses a downloaded web page to automatically cause a client browser to later automatically access the host, and to display a message indicating that the client has preferred access together with a countdown time. As an alternative, a program file can be stored on the client which is effective to launch the client's web browser and to direct access to the particular host at the appointed time, irrespective of whether the client's browser is active.

24 Claims, 5 Drawing Sheets



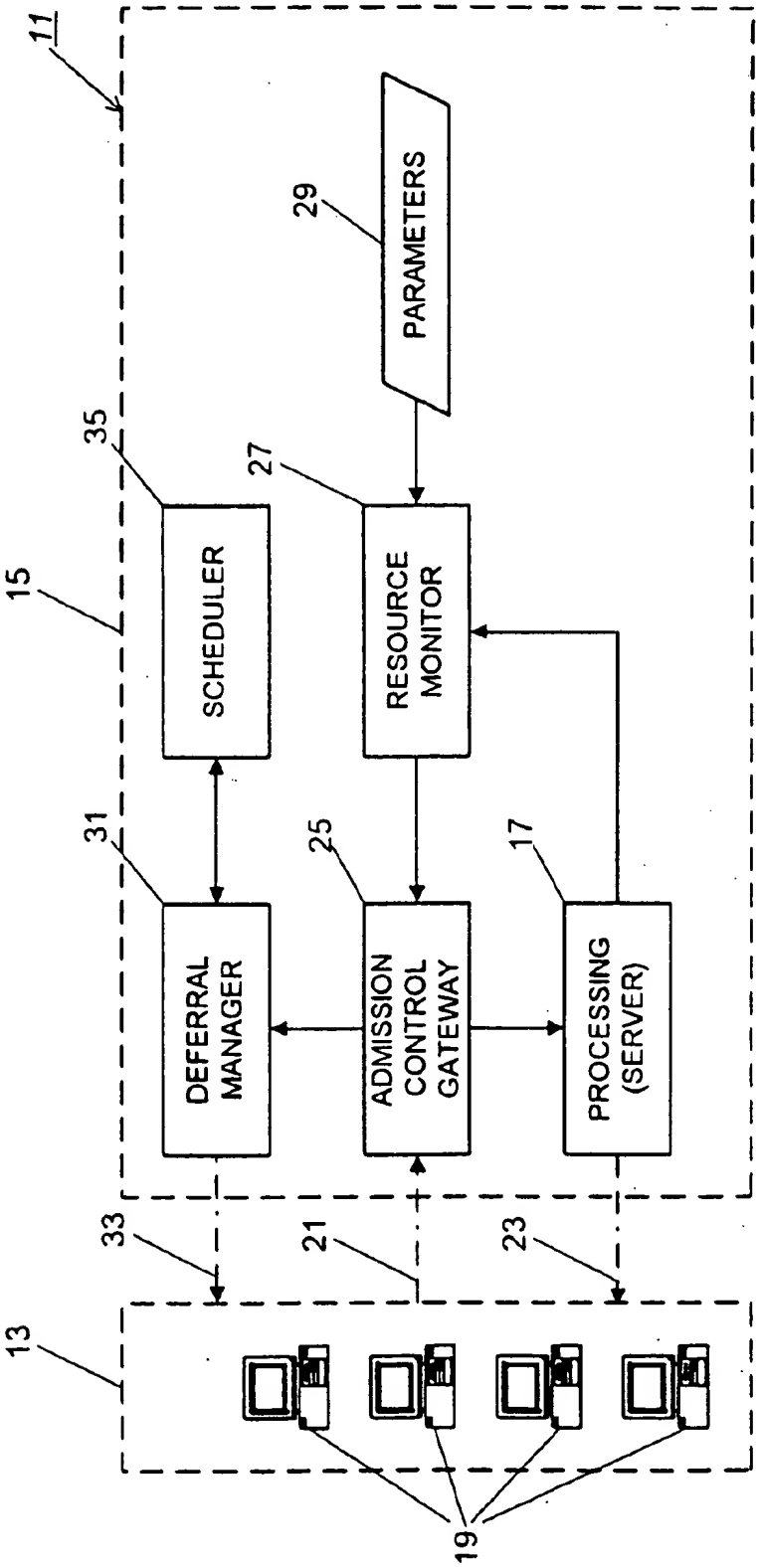

11
29
PARAMETERS
35
27
15
SCHEDULER
RESOURCE MONITOR
31
25
17
DEFERRAL MANAGER
ADMISSION CONTROL GATEWAY
PROCESSING (SERVER)
33
21
23
13
19

FIG. 1

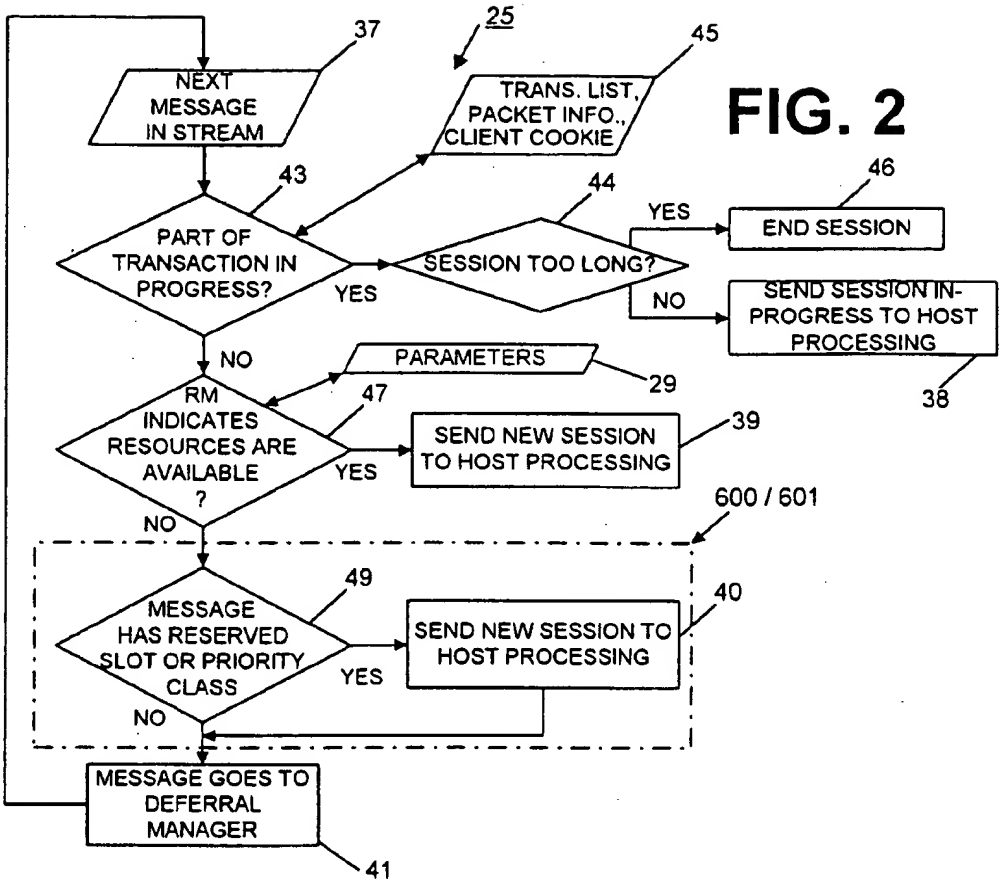

25
37
NEXT MESSAGE IN STREAM
45
TRANS. LIST, PACKET INFO., CLIENT COOKIE
43
PART OF TRANSACTION IN PROGRESS?
YES
44
SESSION TOO LONG?
YES
46
END SESSION
NO
SEND SESSION IN-PROGRESS TO HOST PROCESSING
38
NO
PARAMETERS
29
47
RM INDICATES RESOURCES ARE AVAILABLE ?
YES
SEND NEW SESSION TO HOST PROCESSING
39
NO
600 / 601
49
MESSAGE HAS RESERVED SLOT OR PRIORITY CLASS
YES
40
SEND NEW SESSION TO HOST PROCESSING
NO
MESSAGE GOES TO DEFERRAL MANAGER
41

FIG. 2

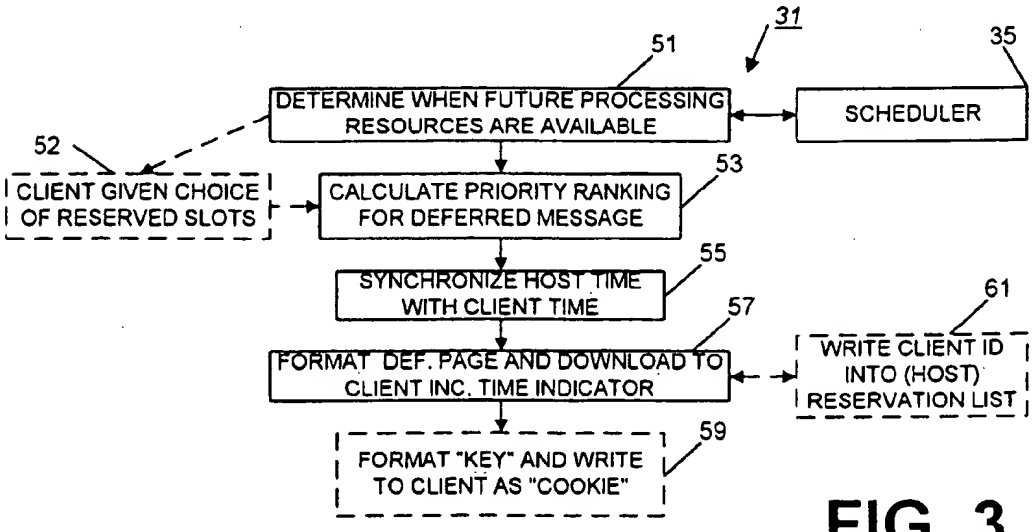

31
51
DETERMINE WHEN FUTURE PROCESSING RESOURCES ARE AVAILABLE
35
SCHEDULER
52
CLIENT GIVEN CHOICE OF RESERVED SLOTS
53
CALCULATE PRIORITY RANKING FOR DEFERRED MESSAGE
55
SYNCHRONIZE HOST TIME WITH CLIENT TIME
57
FORMAT DEF. PAGE AND DOWNLOAD TO CLIENT INC. TIME INDICATOR
61
WRITE CLIENT ID INTO (HOST) RESERVATION LIST
59
FORMAT "KEY" AND WRITE TO CLIENT AS "COOKIE"

FIG. 3

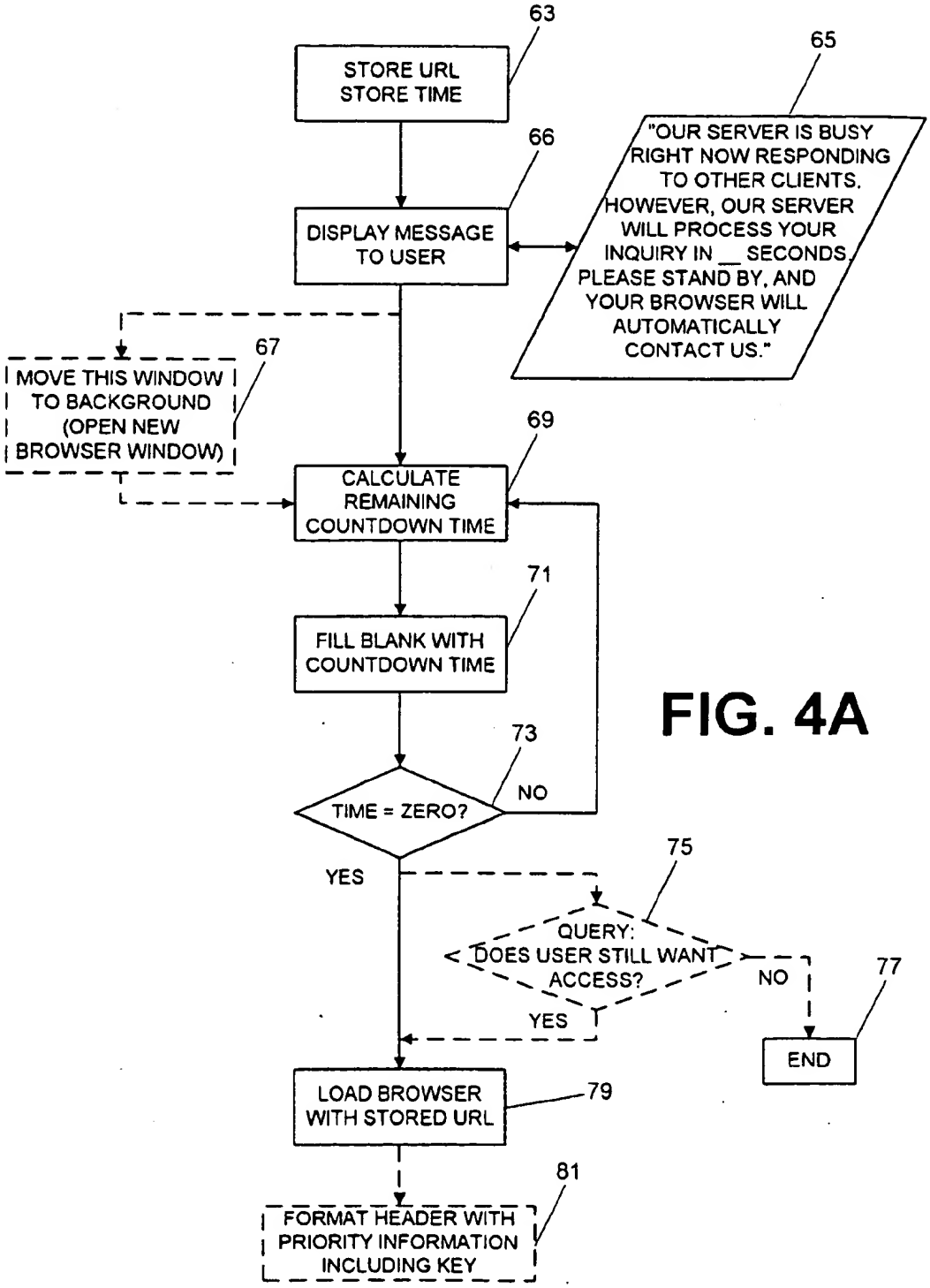

63
STORE URL
STORE TIME
65
"OUR SERVER IS BUSY RIGHT NOW RESPONDING TO OTHER CLIENTS. HOWEVER, OUR SERVER WILL PROCESS YOUR INQUIRY IN __ SECONDS. PLEASE STAND BY, AND YOUR BROWSER WILL AUTOMATICALLY CONTACT US."
66
DISPLAY MESSAGE TO USER
67
MOVE THIS WINDOW TO BACKGROUND (OPEN NEW BROWSER WINDOW)
69
CALCULATE REMAINING COUNTDOWN TIME
71
FILL BLANK WITH COUNTDOWN TIME
73
TIME = ZERO?
NO
YES
75
QUERY: DOES USER STILL WANT ACCESS?
NO
77
END
YES
LOAD BROWSER WITH STORED URL
79
81
FORMAT HEADER WITH PRIORITY INFORMATION INCLUDING KEY

FIG. 4A

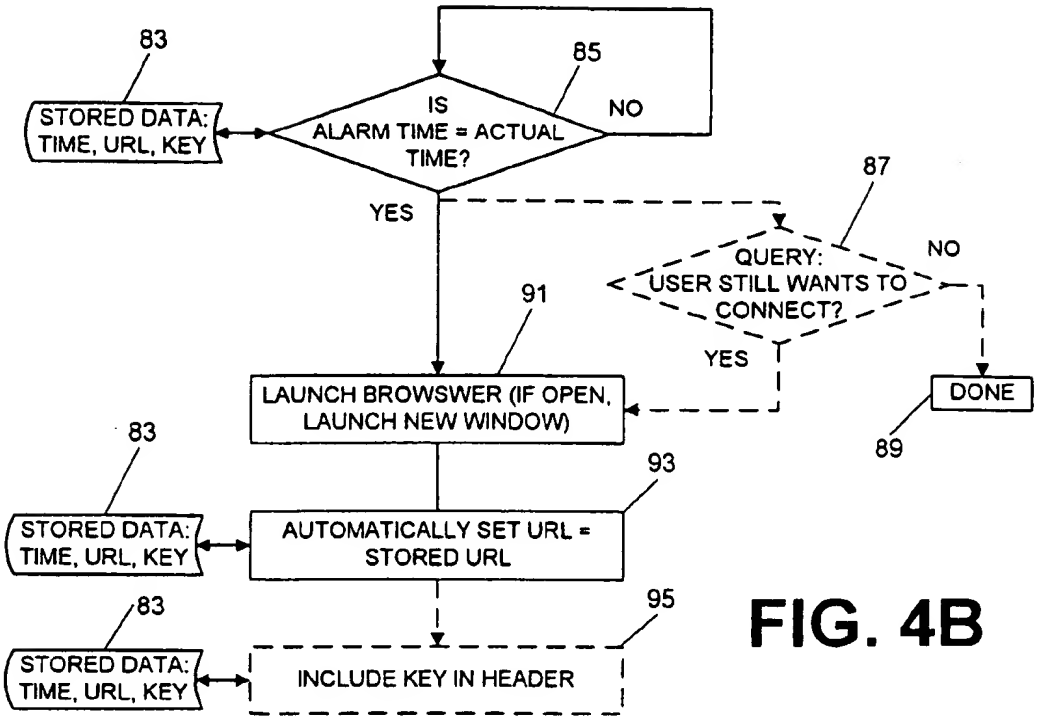

83
STORED DATA: TIME, URL, KEY
85
IS ALARM TIME = ACTUAL TIME?
NO
YES
87
QUERY: USER STILL WANTS TO CONNECT?
NO
YES
DONE
89
91
LAUNCH BROWSWER (IF OPEN, LAUNCH NEW WINDOW)
83
STORED DATA: TIME, URL, KEY
93
AUTOMATICALLY SET URL = STORED URL
83
STORED DATA: TIME, URL, KEY
95
INCLUDE KEY IN HEADER

FIG. 4B

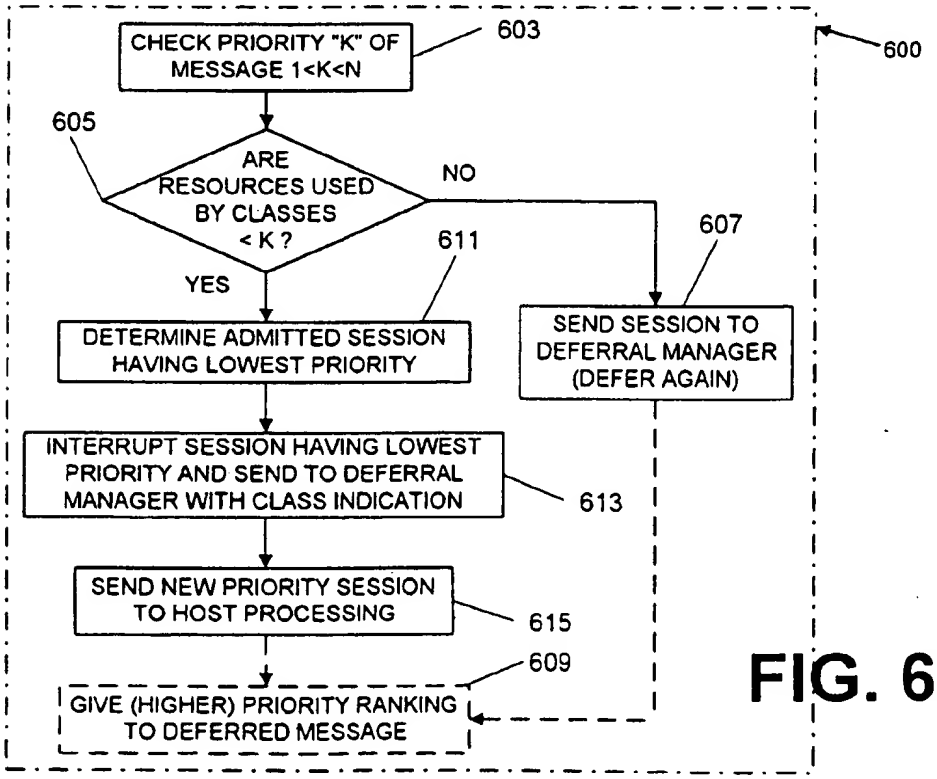

600
CHECK PRIORITY "K" OF MESSAGE 1<K<N
603
605
ARE RESOURCES USED BY CLASSES < K ?
NO
YES
607
611
DETERMINE ADMITTED SESSION HAVING LOWEST PRIORITY
SEND SESSION TO DEFERRAL MANAGER (DEFER AGAIN)
INTERRUPT SESSION HAVING LOWEST PRIORITY AND SEND TO DEFERRAL MANAGER WITH CLASS INDICATION
613
SEND NEW PRIORITY SESSION TO HOST PROCESSING
615
609
GIVE (HIGHER) PRIORITY RANKING TO DEFERRED MESSAGE

FIG. 6A

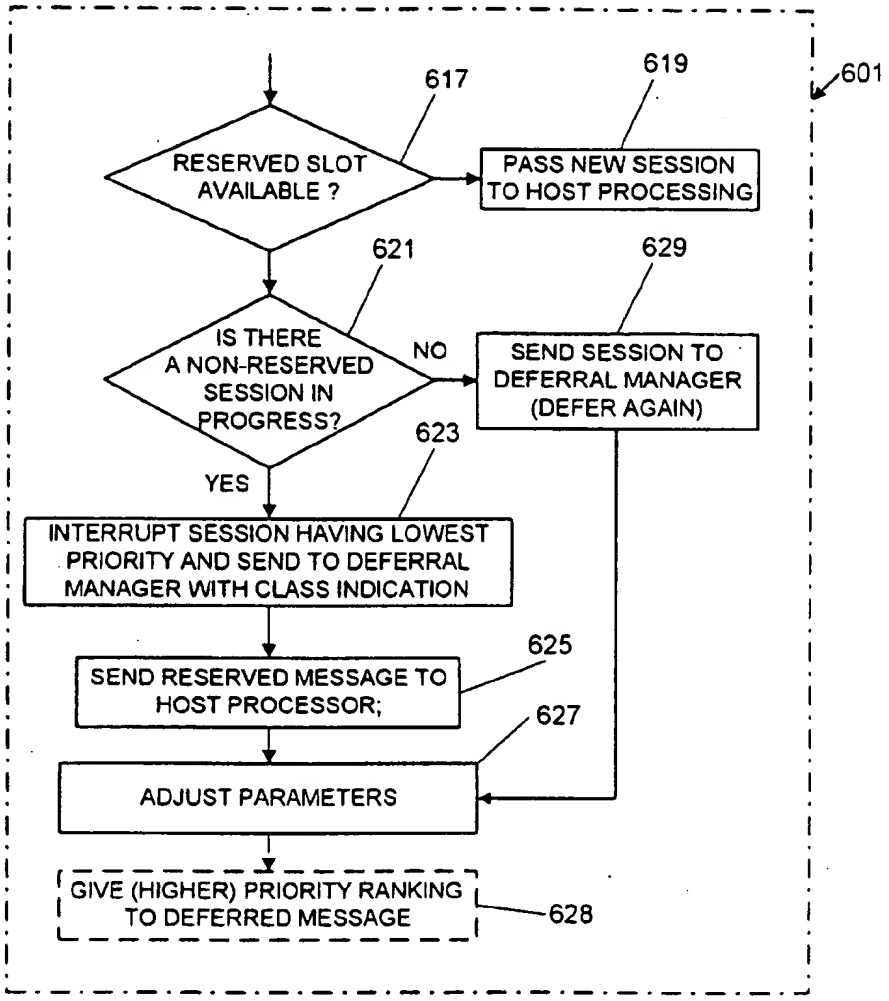

601
617
RESERVED SLOT AVAILABLE ?
619
PASS NEW SESSION TO HOST PROCESSING
621
IS THERE A NON-RESERVED SESSION IN PROGRESS?
NO
629
SEND SESSION TO DEFERRAL MANAGER (DEFER AGAIN)
YES
623
INTERRUPT SESSION HAVING LOWEST PRIORITY AND SEND TO DEFERRAL MANAGER WITH CLASS INDICATION
625
SEND RESERVED MESSAGE TO HOST PROCESSOR;
627
ADJUST PARAMETERS
GIVE (HIGHER) PRIORITY RANKING TO DEFERRED MESSAGE
628

FIG. 6B

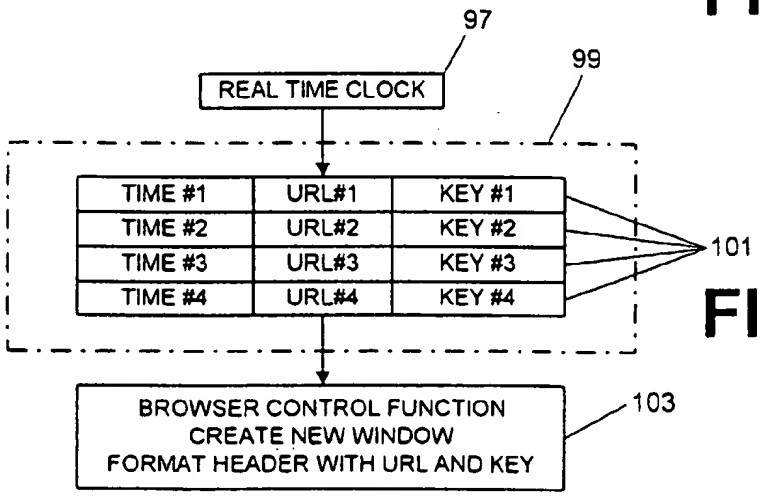

97
REAL TIME CLOCK
99
TIME #1 URL#1 KEY #1
TIME #2 URL#2 KEY #2
TIME #3 URL#3 KEY #3
TIME #4 URL#4 KEY #4
101
BROWSER CONTROL FUNCTION
CREATE NEW WINDOW
FORMAT HEADER WITH URL AND KEY
103

FIG. 5

ADMISSION CONTROL SYSTEM WITH MESSAGES ADMITTED OR DEFERRED FOR RE-SUBMISSION AT A LATER TIME ON A PRIORITY BASIS

The present invention relates to communications between computers and, more particularly, to enhancing quality of service in applications where one computer receives and processes messages from other computers.

BACKGROUND

Several protocols exist in which one computer (a "host") receives and processes messages from a number of other computers ("clients"). For example, in applications involving the world-wide web, a server can receive and process many concurrent messages from different personal computer users; in this example, the server would be the "host" while each personal computer would be a "client."

Messages can usually be grouped into sessions, with each session each having one or more related messages. For example, a multiple-message session could consist of a message requesting information over the world-wide web, and an associated response. Alternatively, a multiple-message session could consist of a commercial transaction, with related messages respectively used to locate a web site for a precise product, submit an order or billing and shipping information, and convey a confirmation of sale to a particular client. Whether a host is to process just a single message or a series of related messages, is usually important to quickly, accurately and completely service each message and each session.

The term "quality of service" refers both a host's ability to provide quick response to a message and to complete an entire session. As a particular host becomes more popular, and due to that popularity receives more messages, the host's processing resources can become stretched. For example, due to heavy traffic, a host may not be able to respond to a message at all, or the host may not provide a timely response (which can cause a client to "time-out" and generate an error). Poor quality of service can have significant results, as users may become frustrated and simply give up trying to reach a particular host, or the sponsor of the host may lose sales or fail to communicate needed information to any or all clients.

Two techniques are generally used to alleviate quality of service problems.

First, more processing capacity can be added to the host, typically by either replacing the host with another, more powerful computer, or by providing multiple computers in parallel and delegating new messages to different ones of the multiple computers. While this first technique presents an effective way of reducing some quality of service problems, it is not always practical. For example, sometimes, due to inadequate planning, budgetary constraints or space constraints, additional processing capacity simply cannot be added. Other times, if demand for a host is not properly forecast, there may be a long lead time before additional processing capacity can be purchased and implemented.

A second technique calls for applying "admission control," where only a certain number of client messages are processed ("admitted") and the remainder are refused; of the messages which are in fact admitted, all are ideally handled in an expedient manner without degradation of quality of service as to those admitted messages. An advantage of this technique is that admission control can be implemented in software, thus facilitating quick, inexpensive use with little

advance notice. Unfortunately, typical admission control mechanisms operate by admitting messages on a message-by-message basis, and so, these typical admission control messages do not provide an adequate solution for multiple-message sessions. Also, the messages which are not admitted to the host are generally not handled at all, such that a client is not informed that the request has been refused or the client, if informed, is simply asked to "try again later." Typically, a refused client must try repeatedly to obtain service with no guarantee that future requests will be processed. For these reasons and others, techniques generally used to alleviate quality of service problems are not always successful.

A definite need exists for an admission control system having an improved ability to alleviate quality of service problems. In particular, a need exists for an admission control system which responds to all messages, whether or not those messages are actually admitted. Ideally, such system would operate by admitting entire sessions, not just individual messages, such that messages relating to a session in-progress are generally admitted. With a system of this type, admission control would at least provide a reliable means of finishing each session with high quality of service. Finally, a need exists for a system that provides some level of service to all clients, including those which have been refused admission. The present invention solves these needs and provides further, related advantages.

SUMMARY

The present invention provides an admission control system both which informs a client of a later time when the client can expect to gain admission to a site and which also provides a means of automatically (or electively) re-submitting a message at the appointed time. The present invention thus facilitates a response to all clients, even deferred clients, to thereby provide better quality of service. Further, however, the preferred admission control system is also session-based, such that once a message is admitted, all related messages (within the session) are processed on a priority basis, such that sessions in-progress can be completed quickly. By facilitating quick completion of admitted sessions, and providing an effective mechanism for handling deferred clients, the present invention provides a low-cost mechanism for significantly enhancing quality of service.

One form of the invention provides an admission control system having an admission control gateway, a deferral manager and a scheduler. When the admission control gateway receives a message that calls for a new client session, the gateway determines whether a processing threshold has been reached; if the threshold has been reached or surpassed, the message is passed to the deferral manager to formulate a response to the particular client. The scheduler is checked to determine a time when the host can expect to have processing resources available, and the deferral manager then formulates a time indication which tells the client when the client can expect to gain admission to the host.

In more detailed features of this first form of the invention, the deferral manager can determine a time for admission based on the use of reserved time slots, which are allocated on a first-come, first-served basis; optionally, a client can be afforded a choice of these slots, to pick a time convenient for the client's user. The deferral manager can then formulate a countdown time and provide the particular client both with the countdown time and a "key" that will enable the client to gain preferred access to the host at

expiration of the countdown time. The key could be a "cookie" or a special password that, when passed to the host upon a retry, guarantees processing on a priority basis. In a different feature of the invention, the deferral manager can operate without specifically reserving time slots, and simply defer sessions in-progress if a priority message is received which would cause the host processor to operate with a greater-than-desired load. For purposes of assigning future times or appointments, the scheduler can operate in several different ways, for example, by setting "appointments" using a maximum number of new sessions per minute, or by monitoring periodic host activity and assigning future sessions when the host is normally "less busy." For example, if regular monitoring reveals that the host is usually not busy between 3:00 and 5:00 O'clock, the deferral manager could tell the client to try again during that time interval.

A second form of the invention provides client resident software for use in accessing a host. Specifically, the client resident software receives a countdown time from a host's deferral manager and manages "when" and "how" the client again attempts access to the host. This second form of the invention can be in the form of a web browser, which automatically maintains one or more countdown buffers or alarms, and which manage repeated attempts at access to the host. Alternatively, this second form of the invention can be a pervasive program which automatically launches a web browser when the appointed time has been reached.

The invention may be better understood by referring to the following detailed description, which should be read in conjunction with the accompanying drawings. The detailed description of a particular preferred embodiment, set out below to enable one to build and use one particular implementation of the invention, is not intended to limit the enumerated claims, but to serve as a particular example thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of a preferred admission control system which implements principles of the present invention. In particular, FIG. **1** at its left shows a plurality of clients which may communicate with a host processing system, seen at the right side of FIG. **1**.

FIG. **2** is a block diagram that indicates operation of an admission control gateway, seen in FIG. **1**.

FIG. **3** is a block diagram that indicates operation of a deferral manager, seen in FIG. **1**.

FIG. **4A** is a flow diagram indicating the operation of a preferred client access mechanism, namely, an exemplary web page which can be formatted by a deferral manager and downloaded together with a special "cookie" to a client. The web page seen in FIG. **4A** implements a countdown timer which, once a predetermined amount of time has elapsed, is used in conjunction with the cookie to gain access to a host which was earlier too busy to process requests from the client.

FIG. **4B** is a flow diagram showing an alternative client access mechanism, where a file is downloaded to a deferred client; the file runs on the clients computer to normally monitor time, and provides an alarm at a time when the deferred client can expect to gain admittance to the host. The file in this embodiment is preferably pervasive, such that it provides an alarm and permits direct launch of a communication mechanism, such as a web browser.

FIG. **5** is a block diagram showing an alternative client access mechanism, namely, a web browser maintains, once deferred by a host, one or more alarm buffers which are used to alert a user when the host is expected to be available.

FIG. **6A** is a flow diagram which indicates operation of a block **600/601** of FIG. **2**, namely, a portion of the preferred admission control gateway which processes or defers new sessions based on assigned class of service and which also assigns class of service.

FIG. **6B** is a flow diagram which indicates alternative operation of the block **600/601** of FIG. **2**, namely, a portion of the admission control gateway which processes or defers new sessions based on the use of contention slots and reserved slots; in this embodiment, the host maintains a variable load threshold used to admit sessions which do not have an appointment. The host changes this threshold to hold more resources in reserve at times when larger volume of reserved access are expected.

DETAILED DESCRIPTION

The invention summarized above and defined by the enumerated claims may be better understood by referring to the following detailed description, which should be read in conjunction with the accompanying drawings. This detailed description of a particular preferred embodiment, set out below to enable one to build and use one particular implementation of the invention, is not intended to limit the enumerated claims, but to serve as a particular example thereof. The particular example set out below is the preferred specific implementation of an admission control system, namely, one which provides priority access based on multiple classes of service, deferral of certain messages, and a web page downloaded to a client having automatic or elective attempts to later attain access. The invention, however, may also be applied to other types of systems as well.

I. INTRODUCTION TO THE PRINCIPAL PARTS

The preferred embodiment is an admission control system resident on a server, a client computer (typically a personal computer), or both. The admission control system may be implemented in firmware, hardware, or software, but most typically will be implemented in software such that it can be optionally implemented on a server which has processing resources which are sometimes strained. The preferred application of the admission control system is to systems involving access and processing on the world-wide web (the "web").

In accordance with the principles of the present invention, the preferred admission control system normally admits messages to a server, but if processing resources of the server are strained, the admission control system defers messages corresponding to new sessions to a later time when it is thought that the server can guarantee processing of the deferred message as a priority message and any corresponding session. The admission control system formats a response to the client, to inform the client's user that access has been deferred, and accords the client a means of later obtaining access on a priority basis if the client contacts the server again at the proper time.

On the client side of the admission control system, the client's user is preferably afforded a means of automatically contacting the server again, once the appointed time has been reached. In the preferred embodiment, the admission control system operates principally on the server and formats a special web page which is downloaded to the client as part of a deferral message. This special web page provides a countdown function, visible to the client's user, which indicates time until re-submission in minutes or seconds; if the client closes the browser or uses it to contact a different

web page, the web page is disabled and the client will not automatically contact the server (in the preferred embodiment). This implementation is preferred, because it can be implemented entirely in software on the server side (including formation of the special web page which is downloaded to the client).

FIG. **1** shows a block diagram illustrating a preferred admission control system **11**. In particular, the diagram illustrates a hypothetical connection between a client side **13** and a host side **15**. The host side includes one or more servers **17**. Notably, it is common to find configurations where several servers are arranged in parallel to share processing responsibility for a single web page (such as a "round-robin" configuration), but for the present discussion it will be assumed that only one server is present; the present invention can be implemented on multiple-server systems as well. The client side **13** may include many individual personal computers **19**, which may each contact the host side **15** via a complicated network of communications paths and nodes. Irrespective of the particular communication path, the host side **15** receives a stream of incoming messages **21** which may access one or more web pages stored on the server **17**, and provides a stream of outgoing response messages **23** in response.

Admission control is effected by passing the incoming messages **21** to an admission control gateway **25**, which determines whether to admit the messages to the server **17**. Messages corresponding to sessions in-progress are preferably always admitted, such that the user of a client can reliably expect a transaction with a server, once initiated, will be completed. It is believed that ensuring completion of sessions in-progress will promote less user frustration, e.g., commercial transactions over the web, once begun, will be reliably completed to satisfaction of the user. At the same time, rather than stretch processing resources of the server, if too many new messages are received, the admission control system defers at least some amount of those messages, such that sessions in-progress can be reliably and quickly completed through the use of un-stretched server resources. A mechanism for responding to deferred clients preferably includes an affirmative statement about admittance, e.g., that the client will later be admitted at a time specific, to avoid too much frustration on the part of users of deferred clients.

The admission control system of the preferred embodiment includes a resource monitor **27**, which provides information on current use of server resources. The resource monitor is typically a set of code that uses a standard call function provided by most server protocols; the call function returns a value having a particular format, depending upon the server. For example, some servers will return a percentage, indicating present resources which are "occupied," while other servers may return an numeric value which is not based on a decimal scale (e.g., not a number from one to ten or one to one-hundred); Other servers may return a counter which needs to be polled periodically to determine resource utilization.

A message received by the admission control gateway **25** is first analyzed to determine whether it corresponds to a session in-progress. If a session has already been established (and the newly-received message is a part of a continuing transaction between the server and a particular client), then the message is preferably admitted to the server. If not, the resource monitor **27** can be used to determine whether the message is to be admitted as the start of a new session. To this effect, the resource monitor **27** is coupled to a set of one or more parameters **29** which are stored in a buffer. These parameters define a desired maximum load (or threshold for deferral) which may be static or may vary, depending upon the embodiment of admission control. For example, one embodiment discussed below uses reserved time slots which can be allocated to deferred messages; if a large volume of deferred messages are scheduled for processing on a priority basis at three O'clock, a stored parameter can be set for three O'clock to define a lower threshold, such that more messages (which are non-priority messages) are deferred than normal at that time. In the preferred embodiment, a static figure is used (e.g., eighty percent), primarily for simplicity in implementation. The admission control gateway **25** compares the set of at least one parameter with the result of the call function and, if the comparison indicates that server processing resources have exceeded the threshold, then new sessions not having priority are deferred.

Since the admission control gateway **25** defers some messages corresponding to new sessions when resources are stretched, it is desired to provide some reply to a client system which submitted the message, in order that the user of the client system will not become frustrated or continually re-submit the message (thereby further overloading server resources). To accomplish this end, the admission control system **11** further includes a deferral manager **31**, which formats and provides a response message **33** to the client system which submitted the deferred message. Preferably, the deferral manager **31** is coupled to a scheduler **35** which, together with the deferral manager, calculates a later time when it can be expected that the deferred message can be processed by the server **17**. The scheduler has many possible implementations for obtaining such information. For example, the scheduler can compile statistics based on day-to-day operation of the server and times when the processing resources of the server tend to be less strained; in this example, the scheduler could determine that a particular server is "less busy" from twelve O'clock noon until one O'clock P.M., and could defer a client system until twelve O'clock noon and the one hour time range thereafter. Alternatively, the scheduler could simply set "appointments" (e.g., two for every five minutes) and simply return to the deferral manager **31** a time for the next available appointment. In the preferred embodiment, the scheduler uses the latter function and defers messages for at least a minimum predetermined amount of time, e.g., 300 seconds or more as indicated by Table I, below; in conjunction with a time set by a web page which is downloaded to the client, the client's message is later accepted on a priority basis if the client contacts the server within a defined interval following the time. Implementation of the scheduler is effected in the preferred embodiment via software.

A number of mechanisms can also be implemented such that the admission control system **11** recognizes a deferred message as a priority message following re-submission. In the preferred embodiment, the deferral manager **31** generates a "key" in the form of a "cookie" which the admission control system writes to a hard disk of the client system. When access is again requested by the client system, the admission control gateway **25** interrogates the client system to determine whether the cookie is present and, if so, the admission control gateway accords priority status to a message from the client system at that time, in terms of generating a new session. In addition to writing the cookie to the client system, the deferral manager **31** also generates an informative web page which it downloads to the client's browser. This web page visually displays to a user of the client system an informative message, e.g.,

> "We're sorry, but our server is temporarily serving other clients; to better assist you, we have scheduled an

appointment for your transaction, and if you do not exit this web page, your browser will automatically contact us in 23 seconds."

The numeric figure in the above message is a countdown time, and the web page generated by the deferral manager automatically establishes a countdown mechanism on the client system. The countdown time is continually displayed to a user, and once the time reaches zero, the web page automatically directs the browser to the same URL which resulted in the deferral, and the admission control gateway **25** checks for the presence of the aforementioned cookie.

Certain alternative implementations are also discussed below, where a web page is not used; these embodiments include systems where the browser is a modified browser on the client side, which includes a buffer (not visible to the client's user) for storing a uniform resource locator ("URL") of a server which has deferred access, together with an appointment time. The modified browser is effective to (in the "background" as far as concerns the user) automatically detect when the appointed time has been reached and then cause the browser to submit a priority message for processing by the client. Ideally, the user is afforded in this embodiment an election at some point during the process, as to whether the user wishes to establish deferred access. The browser is then directed to automatically open up a separate window for each new session initialized via the buffer. Ideally also, the buffer of the modified browser can store indications for several deferred, priority sessions, and the modified browser is effective to order such indications such that they are used to establish deferred sessions at the proper times, regardless of the order in which the indications are received.

It should be understood that the preferred application is to Hypertext Transfer Protocol ("HTTP"), but that there are many applications for the admission control system of the present invention. For example, common protocols to which admission control can be applied include Internet Protocol, User Datagram Protocol, Simple Mail Transfer Protocol, Network File System Protocol, applications on local area networks, and other communications protocols where a host services one or more client systems.

With the principle features of the preferred embodiment (and some of the alternative embodiments) introduced, implementation of the preferred embodiment will now be further described.

II. SERVER SIDE ADMISSION CONTROL

FIGS. **2** and **3** are used to illustrate operation of the admission control on the server side in additional detail. In particular, FIG. **2** shows additional detail for the operation of the admission control gateway **25**, whereas FIG. **3** shows additional detail regarding the operation of the deferral manager **31**.

As seen in FIG. **2**, the admission control gateway **25** processes each next message **37** in the stream by either sending the message to host processing (e.g., to the server), as indicated by the reference numerals **38**, **39** and **40**, or by sending the message to the deferral manager, as indicated by the reference numeral **41**.

The admission control gateway first interrogates each message to determine whether the message is part of a session in-progress, as indicated by a software block **43**. Several different types of data can be used for this purpose, as indicated by a data block **45**. First, the admission control gateway can maintain a transaction list which includes information on the requesting client and the session at-issue; the admission control gateway monitors header information for each message and simply determines whether information corresponds to a list of sessions in-progress. Second, the admission control gateway can simply look for a particular password or identifier provided with the packet information for the message. Third, the admission control gateway can also look for a client cookie which has been written to the requesting client, by simply interrogating the client via a response message to determine whether the cookie is present. Of these processes, a transaction list is preferably used as the most expedient of these processes.

For sessions which are in progress, the admission control gateway **25** applies criteria to limit overall session length (as represented by reference numeral **44** in FIG. **2**), so as to ensure that all sessions are terminated as of some point in time. For example, as seen in Table I (below), session length may be capped to 1800 seconds, with the admission control gateway invalidating any session which exceeds that length of time, as designated by the reference numeral **46**. When sessions are admitted, they are assigned to a class, and the class identifier is included in a cookie stored on the associated client system. The server or admission control system keeps statistics for each request processed, for example, statistics on the last 1000 requests (for each class, if multiple classes of priority are used). Typically, a server will not need to terminate sessions, but it (via the admission control system) will simply stop admitting sessions in an over-represented class and allow natural attrition to reduce its population. If this does not occur quickly enough then sessions are terminated by invalidating their session cookie. In each case where a cookie is used, alternative state tracking mechanisms can be used. The server could have a table of the states of the current sessions and simply update or read entries from that table. Sessions which have not been in-progress for an excessive length of time are passed along for processing, as indicated by the block **38**.

If the message is not part of a transaction in-progress, the admission control gateway **25** can also admit the message as a new session if it is determined that the server has sufficient processing resources available, as determined by comparison with the set of parameters **29** (indicated in FIG. **2** by the reference numeral **47**). The admission control gateway **25** can further admit the message if it is determined that the message corresponds to a message which has been previously deferred, as indicated by a decision block **49**. In this regard, these two determinations can be performed in alternative order depending upon the particular implementation. For example, in an embodiment where reserved time slots are used and dynamic parameters **29** are used to regulate un-reserved (or contention) time slots, it is preferable to detect new sessions having reserved time slots so as to not reduce the number of contention slots available. In the preferred embodiment, specific slots are not used and the parameters **29** include a single, static threshold, with lowest priority (e.g., most recently received, non-priority) messages being deferred, even if already in-progress, if the server only has processing resources for priority messages. Consequently, the determinations can equivalently be performed in the preferred embodiment in the order indicated in FIG. **2**. Notably, processing using reserved time slots or priority classes is identified as a processing block by the reference numeral **500/501**, and will be discussed further below. If the determinations do not result in admission of the message to the server, then the message is sent to the deferral manager, as indicated by the bottom most block **41** of FIG. **2**.

The operation of the deferral manager **31** is indicated in FIG. **3**, and results in the sending of a deferral message to the

client system, and the creation of a priority identifier that permits re-submission of the deferred message on a priority basis. In the preferred embodiment, the deferral manager operates in communication with the scheduler **35** to determine a time when future processing resources of the server will be available on a priority basis for deferred messages (as indicated by a functional block **51** of FIG. **3**). The code which implements the deferral manager then proceeds to calculate a priority ranking for the deferred message, as indicated by block **53** of FIG. **3**. Optionally, if reserved time slots are used, the admission control system can give the client system a choice of reserved time slots, as indicated by the block **52** of FIG. **3**; this function can either be done by permitting the user of a client system to select from available time slots, or by taking a preferred client time and if slots are unavailable, requiring the client to choose from other options. In the preferred embodiment only two classes of service are used, namely, for priority messages and for messages not having priority status; additional distinctions are made based upon newest session (i.e., a priority message representing the most recently commenced priority session is viewed as less important than priority sessions in-progress, and a non-priority message corresponding to the most recently commenced non-priority session is viewed as less important than other non-priority sessions in-progress). As a consequence, block **53** operates in the preferred embodiment by simply ascribing priority to any deferred message.

Importantly, in one preferred embodiment, a number of classes of service can be made available for messages and different web pages, each having different priority ranking (1 to n). For example, if a server is used to support the web pages of two companies, and one company pays a premium fee for support of a first web page, then messages directed to the first web page can be given priority when the server's overall processes are strained, priority including preferred access over messages directed to the second web page. Alternatively, if processing resources become very stretched and reserved time slots are not used, it is possible that a server could at a particular point in time be processing only priority messages, in which case a message representing new session could in some embodiments be deferred twice; using multiple classes of service, the again-deferred message could receive a second step upward in priority, and so on. Additional details regarding applications using multiple classes of service are discussed in U.S. patent application Ser. No. 09/038,657, for "Host Processing With Multiple Classes of Service," filed on the same date as this disclosure by inventor Peter Phaal, which is hereby incorporated by reference in its entirety, as though set forth herein.

The deferral manager **31** then proceeds with the optional step of synchronizing host time with client time (indicated by block **55**). This process can be as simple as calculating a countdown time in seconds and downloading that information to the client system (as is done in the preferred embodiment), or in alternative implementations, the deferral manager can interrogate the client system to determine differences in time, and adjust the time indicator sent to the client accordingly. As indicated by block **57** of FIG. **3**, the deferral manager **31** formats an informative message including the time indicator and sends this information to the client system, preferably including a message having a purpose to avoid user frustration (e.g., a message indicating that the client's request will be processed at a specific time, with automatic re-submission by the client's web browser). Preferably, the informative message is contained within a deferral web page which stores the URL of the server for which deferral occurred, and the deferral web page again submits the same URL at expiration of a countdown time.

The deferral manager **31** also stores an identifier of priority status for each deferred message, via one of two alternative mechanisms (each indicated by dashed lines in FIG. **3**). First, the identifier can be formatted as a "key," such as a password or cookie written to the client system (indicated by reference numeral **59**) which identifies priority status. Alternatively, the deferral manager can (as indicated by block **61** of FIG. **3**) write the identifier into memory on the server side of the admission control system, as part of a list. Use of a cookie is preferred, since such an implementation minimizes the amount of processing time required of the deferral manager, hence provides for greater throughput in processing deferred messages if a large number of deferrals are occurring. The cookie should include a unique identifier for the particular server (or web page), and it should also be time-limited; the cookie can either include the time indicator itself (i.e., a cookie which varies from message-to-message), or it could be a unique value that changes each day, for example.

A. SCHEDULER OPERATIONS.

The scheduler may operate in any of a number of ways in order to optimize server processing. Preferably, the user of a client system should not have to wait long prior to re-submission of a message and, consequently, the preferred embodiment allocates appointments up to a predetermined number of deferred sessions per minute, and assigns messages as soon as possible after deferral. This predetermined number of sessions can be varied using a server side configuration screen [indicated in Table I, below]. The configuration screen also specifies an exclusion time during which is the minimum amount of time following deferral for a corresponding appointment.

In an example indicated in Table I, below, the minimum time before an appointment is indicated to be 300 seconds, with a maximum number of deferred sessions indicates as ten sessions per minute. In this example, if a total of 22 sessions were to be deferred at approximately 3:00 P.M. (and there was no backlog of deferred messages), the deferral manager **31** would provide priority to the first ten messages if they re-submitted their messages between 3:05 P.M. and 3:06 P.M., the next ten message between 3:06 P.M. and 3:07 P.M., and the final two messages between 3:07 P.M. and 3:08 P.M. If another message received at 3:01 P.M. was to be deferred, the message would receive priority between 3:07 P.M. and 3:08 P.M., together with the 21st and 22d messages referred to above. It the message were instead received at 3:04 P.M. (instead of at 3:02 P.M.), the deferral manager and scheduler would assign priority to the message between 3:09 P.M. and 3:10 P.M., even if there were no backlog of messages.

TABLE I

(SAMPLE ADMISSION CONTROL CONFIGURATION SCREEN)

[About | Index | Statistics]
Web Flow Settings
Admit Sessions:
[ ] Shutdown in [600] seconds.
[ ] Always (disables dynamic admission control).
[X] If Load < [5.00]
End Sessions When:
[ ] interval between requests > [600] seconds.
[X] or, session duration > [1800] seconds.
Respond to Rejected Sessions with:
[ ] a redirect to [ ] (leave blank if no redirection required),

TABLE I-continued

(SAMPLE ADMISSION CONTROL CONFIGURATION SCREEN)
and then if redirected request cannot be accepted or redirection is disabled, respond with:
[ ] Error
[X] Page [count_down.htm]
[X] Allocate up to [10] sessions per minute.
[ ] External
and then exclude them for [300] seconds.

As further indicated by Table I, the server side configuration screen provides for adjustment of maximum load prior to deferral, for admission control to be turned "off," and also permits a number of other options in controlling admission.

III. CLIENT SIDE ADMISSION CONTROL

Client side admission control is preferably achieved automatically by the admission control system, using the deferral manager to format and download to the client a web page that automatically redirects the client's browser back to the particular server when the proper time has been reached. Presently contemplated alternative implementations of client side admission control include: a pervasive program file, which launches either the client's web browser (if the client's browser has been such down) or an additional browser window (if the client's browser is active); and, a modified web browser which automatically stores several appointments for deferred server access. Each of these implementations will be discussed below, with reference to FIGS. **6**A, **6**B and **5**, respectively.

A. CONFIGURATION OF DOWNLOADED DEFERRAL WEB PAGE.

FIG. **4**A presents a flow diagram for the web page which is preferably formatted by the deferral manager **31** and sent to the client system. As indicated by FIG. **4**A, the web page operates as a program and includes, in addition to HTML format instructions downloaded to the client system, of a URL and time indicator which are stored in random access memory ("RAM") of the client system, as indicated by the top-most reference block **63** of FIG. **4**A. The HTML format instructions cause the client system to continually display a text message to the client's user substantially in the format indicated by data block **65** of FIG. **4**A. Preferably, the text message includes a positive statement indicating that the client's request for access has been deferred but that alerts the user that the request will be automatically resubmitted at a specific time.

As with other web pages, the deferral web page is actually downloaded to memory of the client's system, and the browser will continue (as indicated by block **66**) to display the page until the user either attempts to access a different URL or shuts-down the browser. In an optional implementation (indicated by the function block **67**, represented in dashed lines), the web page could direct the browser to open a new window, such that the user can access other URLs while one window continues to display the deferral web page.

As indicated by blocks **69**, **71** and **73** of FIG. **4**A, the deferral web page preferably continues to calculate countdown time and to replace time remaining in the text message with a current countdown time; the web page with each decrement checks the countdown time to determine whether it has reached zero, and if so, proceeds to automatically attempt to access the same URL which earlier resulted in deferral. An optional dialog box can be presented to the user of the client system when the countdown time reaches zero, querying whether the user still wants to access the same web page which earlier resulted in deferral, all as indicated in dashed lines by the box **75** of FIG. **4**A. If the user no longer wishes access, the operation of the web page (e.g., the countdown function) ends, as indicated by block **77** of FIG. **4**A. Whether access is automatic or elective with the user, the web page proceeds to re-submit the same URL which originally resulted in deferral, as indicated by block **79** of FIG. **4**A. If priority is represented by a cookie stored on the client system (or with a password), the web page can be configured to automatically load header information for the URL request with priority information, as indicated by block **81**. This function **81** is indicated as optional (i.e., by dashed lines in FIG. **4**A), because as mentioned earlier, it is preferred to use interrogation of the cookie on the hard drive of the client system by the admission control gateway to resolve priority.

B. DOWNLOADED OF PERVASIVE PROGRAM TO CLIENT.

Particularly in applications where session length is long, or where a long time (i.e., greater than 15 minutes) is required prior to priority access to the server, it may be desirable to download a program file to the client's hard disk for operation outside of the client's browser. Such a program file could be implemented in java script or as a batch file.

As indicated by the data block **83** in FIG. **4**B, such a program file would include data representing a time (synchronized with host time in connection with download by the deferral manager), together with a URL for the web page which resulted in deferral and a key or cookie, if appropriate; the key or cookie could also be stored separately on the client's hard drive as a data file, or on the server in the form of a list. The program file operates to establish an alarm function **85** which continually (i.e., repeatedly and frequently) checks time using a real time clock of the client system to determine time for access to the server. If an alarm is sounded, the program file may optionally query the user of the client system whether access is still desired, as indicated by the function block **87**, and if access is not desired, the program may end, as represented by the function block **89**. Whether or not access is automatic or elective, the program file opens up a new browser window (or launches a first browser window, as indicated by block **91**) and automatically directs the browser to the web site which earlier resulted in deferral. During this process, as indicated by the block **93**, the browser accesses the stored data **83** including the URL. If appropriate to the particular implementation, the program file can also use the stored data **83** to format header information as appropriate, as indicated by the function block **95** of FIG. **4**B.

C. USE OF A MODIFIED WEB BROWSER.

Use of a web browser that is specially adapted for use in the present invention is another embodiment, albeit not the preferred embodiment because it cannot be implemented solely by software operating on the server side. Indicated by FIG. **5**, such a browser should be in with a real time clock **97**, and use information provided by the clock to access a buffer **99**. This buffer, which is implemented in RAM on the client system, has memory locations **101** directed to each URL for which an appointment has been scheduled, together with an appointment time and, as appropriate to the embodiment, a key enabling priority access. At least one such memory location is used by the browser software, and preferably at least four memory locations are used. Th the browser automatically implements an alarm for a deferred message at the proper time, not withstanding the order in

which deferral messages are received from a deferral manager. The web browser preferably includes a function that recognizes a deferral message, and which automatically retrieves each URL and corresponding appointment time for storage in the buffer. Alternatively, the deferral manager can include software for recognizing a browser which supports appointments for deferred messages, and which automatically writes the appropriate URL and appointment time into the appropriate memory location **101**. Once an alarm is reached, the browser automatically creates a new window dedicated to the URL which originally resulted in deferral, and includes header information (including a key) as appropriate in a message requesting access, all as indicated by the block **103** in FIG. **5**.

Modification of a web browser to support the above-recited functionality is considered within the skill of one familiar with software design, and could be added in straightforward manner to source code for common browser programs, such as programs made by "Microsoft" and "Netscape."

IV. USE OF MULTIPLE CLASSES OF SERVICE AND OF CONTENTION AND RESERVED TIME SLOTS

FIGS. **6**A and **6**B respectively indicate operation of the admission control gateway **25** in the case of multiple classes of service and in the case of contention and reserved time slots. These two embodiments differ from one another primarily in the fact that reserved time slots are "held open," e.g., the set of parameters that are used to define maximum desired processing load are made to be dynamic, so as to limit processing of contention messages (that is to say, sessions not having a reserved time slot) when the server is awaiting sessions having a reserved time slot.

FIG. **6**A provides an expanded view of block **600/601** seen in FIG. **2** and labeled as block **600** in FIG. **6**A. The admission control gateway first checks the priority "k" of a message, which can vary from one to "n." The first processing block **603** of FIG. **6**A returns the value "k" for a determination of whether the server has resources available to process the message as a new session.

The motivation for multiple classes can vary: for example, the admission control system can be designed to support multiple deferrals with a higher priority in admission given to those messages deferred more than one (such that messages deferred twice would be given priority over messages deferred once); the multiple classes can also be designed to support one or more web pages, where perhaps one web page provider is paying for a premium service for better service than another web page provider.

A decision function **605** determines whether processing resources are used by classes having priority less than "k." In the preferred embodiment (where only two classes are used), this inquiry is resolved by determining whether non-priority sessions are in-progress. It will be recalled that a block discussed above in connection with FIG. **2** determines whether processing resources are available and admits messages on that basis; consequently, when operation of the admission control gateway reaches the block **605**, it has already been determined that the server has already exceeded its maximum load, and that no more processing resources are available. If the decision function determines that the newly-received message represents the lowest priority session, then the message and its corresponding session are again deferred as indicated by function block **607**; optionally, the embodiment of FIG. **6**A can again increase the priority of a message to "k+1" as indicated by a function block **609** seen at the bottom of FIG. **6**A. If, on the other hand, there is a lower priority session in-progress, the admission control gateway identifies this session in-progress and interrupts that session, sending it to the deferral manager to make room for the new, higher priority message (as designated by numerals **611** and **613**, respectively). The newly-received, higher priority message is then admitted to the system as indicated by function block **615**, while the deferral manager formats a deferral message corresponding to the lowest priority message which was deferred, and assigns it a higher priority and a later appointment.

Use of contention and reserved time slots is illustrated with reference to FIG. **6**B, in which processing functions bear the group reference **601**, corresponding to the block **600/601** from FIG. **2**. As was mentioned earlier, if reserved and contention time slots are used, messages having an appointment should not be placed into contention time slots, and thereby overly limit host processing; the processing indicated by FIG. **2** should have a mechanism that achieves this function. Alternatively, entry into processing **601** represented by FIG. **6**B could be performed prior to a determination of whether the server has spare processing resources which are available.

A first decision block **617** of FIG. **6**B determines whether there is a reserved slot which is available for a message, and messages satisfying this criteria are admitted to the server, as indicated by reference numeral **619**. If the reserved slots are all taken (for example, processing for a prior priority session has taken an excessive amount of time), then a second decision block **621** is called upon to determine whether there are lower priority sessions in-progress. The lowest priority session in-progress can be interrupted and sent to the deferral manager (with an increase in priority for deferred handling), all as indicated by blocks **623**, **625** and **628** of FIG. **6**B. On the other hand, if no lower priority sessions are available, then the newly-received message must be assigned another reserved time slot and is sent to the deferral manager, as indicated by dashed-line processing block **629**. The system can then proceed to adjust parameters for future times, corresponding to future slot reservation, as indicated by function block **627**.

The use of contention and reserved slots can be implemented in situations where average session length is fairly long, greater perhaps than a few minutes. In such cases, it may be desired to limit acceptance of new, contention messages, if the admission control gateway is expecting a large number of new sessions having appointments.

In view of the foregoing description, various alternative embodiments of the present invention will occur to those having skill in electronics. For example, various software alternatives will also occur to those having programming skill which effects deferral of messages without departing from the spirit of the present invention. The admission control gateway, deferral manager and scheduler, above, are typically implemented with a single set of code having various arrangements of routines and subroutines, but some of these functions could also be implemented in hardware or firmware. Other mechanisms for assisting admission control on the client side may be used instead of the mechanisms described above.

Having thus described several exemplary implementations of the invention, it will be apparent that various alterations, modifications, and improvements will readily occur to those skilled in the art. Such alterations, modifications, and improvements, though not expressly

described above, are nonetheless intended and implied to be within the spirit and scope of the invention. Accordingly, the foregoing discussion is intended to be illustrative only; the invention is limited and defined only by the following claims and equivalents thereto.

I claim:

1. An admission control system that selectively admits messages of a stream of messages from at least one client system to a host, comprising:

a resource monitor coupled to the host that measures processing resources of the host, the resource monitor providing an indication of processing resources available to the host;

a deferral manager that receives deferred messages and that responsively determines a time when deferred messages can be processed by the host; and

an admission control gateway that receives both the stream of message and the indication from the resource monitor, the admission control gateway

determining whether each message in the stream corresponds to a session in-progress and, if a message corresponds to a session in-progress, responsively admitting the message to the host,

determining whether messages from the stream not corresponding to a session in-progress can nevertheless be processed by the host, including for at least some messages in the stream, comparing the indication with a threshold, and if a result of comparing indicates additional processing capacity within the threshold, admitting the at least some messages as new sessions, and

if a particular message does not correspond to a session in-progress and a result of determining does not indicate that the particular message can be processed by the host, then sending the particular message to the deferral manager as a deferred message;

wherein the deferral manager sends a deferral message to a client system corresponding to a deferred message which includes an indication of a time when the deferred message can be processed by the host.

2. An admission control system according to claim 1, wherein determining whether messages from the stream not corresponding to a session in-progress can nevertheless be processed by the host includes:

determining whether a message in the stream is a priority message and admitting the priority message to the host processor.

3. An admission control system according to claim 2, wherein:

the deferral manager formats the message sent to the client system to include a key indicating priority status; and

determining whether a message in the stream is a priority message includes receiving a key from the client system indicating priority status.

4. An admission control system according to claim 2, wherein:

the deferral manager causes storage in said admission control system of a client system indicator; and

determining whether a message in the stream is a priority message includes determining whether there is a corresponding client system indicator which has been stored in said admission control system.

5. An admission control system according to claim 1, wherein the host processor is a web server.

6. An admission control system according to claim 1, wherein:

said admission control system further comprises a scheduler which, together with the deferral manager, reserves time slots for future access corresponding to the deferred messages;

the time is a specific time; and

determining whether messages from the stream not corresponding to a session in-progress can nevertheless be processed by the host includes

determining whether a message in the stream corresponds to a reserved slot within a specific range of time that depends upon the specific time.

7. An admission control system according to claim 6, wherein at least one of the scheduler and deferral manager causes the threshold to be varied at a future time to limit processing of messages in unreserved time slots.

8. An admission control system according to claim 6, wherein the scheduler assigns reserved time slots by interacting with the client system to give the client system a choice of time slots.

9. An admission control system according to claim 1, wherein:

said admission control system further comprises a scheduler which, together with the deferral manager, reserves time slots for future access corresponding to the deferred messages;

the time is a range of specific time; and

determining whether messages from the stream not corresponding to a session in-progress can nevertheless be processed by the host includes

determining whether a message in the stream has an associated identifier which corresponds to a reserved slot within a specific range of time dependent upon the range of specific time.

10. An admission control system according to claim 1, wherein the deferral message is formatted by the deferral manager to include a key which must be presented upon repeated attempt to access the host processor in order to receive priority processing.

11. An admission control system according to claim 10, wherein the deferral manager transmits the deferral message to the client system in such a manner that it writes the key as a cookie to the client system.

12. An admission control system according to claim 10, wherein the deferral message is formatted to pass the key as a password to a web browser of the client system.

13. An admission control system according to claim 1, wherein the deferral message includes a program file adapted to cause the client system to resend a message to the host processor at the time when the deferred message can be processed by the host, irrespective of whether a web browser is active upon the client system.

14. An admission control system according to claim 1, wherein the deferral message includes a web page having a countdown time, the web page adapted to cause a browser of the client system to resend a message to the host processor at the time when the deferred message can be processed by the host.

15. An admission control system according to claim 14, wherein the web page is adapted to cause a browser of the client system to resend a message to the host processor at the time in a manner which is elective to a user of the client system.

16. An admission control system according to claim 1, further comprising means for determining, for each message which has priority access, whether sufficient resources of the host processor are available, and if insufficient resources of

the host processor are available, for also interrupting and deferring a session of lower priority in-progress to make room for a session of higher priority.

17. An admission control system according to claim 1, further comprising means for ending a session that has been in-progress for greater than at least a predetermined amount of time.

18. An improvement in an admission control system that selectively admits and defers incoming messages sent from at least one client system, the admission control system consisting of at least one of software, firmware and hardware to effect admission control for a server, said improvement comprising:

determining whether incoming messages correspond to a session in-progress and admitting to the server those incoming messages which do correspond to a session in-progress;

for incoming messages not corresponding to a session in-progress, comparing current server processing resources with at least one predetermined parameter and responsively determining whether a session corresponding to the message can be processed without burdening server processing resources beyond an amount indicated by a predetermined parameter;

for incoming messages not corresponding to a session in-progress,

determining whether the message has been previously deferred by said admission control system, and

if the message has been previously deferred by said admission control system, then admitting the message as a new session;

deferring incoming messages which do not correspond to a session in-progress, have not been previously deferred by said admission control system and cannot be otherwise processed without burdening server resources beyond the amount indicated by a predetermined parameter, including

assigning a time indicator to those messages which are deferred for later admission to the server on a priority basis, and transmitting an indication of deferral and the time indicator to the client system, and

creating a priority indicator associated with deferred messages of prior deferral by said admission control system, the priority indicator adapted for use by said admission control system in determining whether a corresponding message has been previously deferred.

19. An improvement according to claim 18, further comprising:

automatically causing a client system to re-submit a deferred message to a host at substantially the later time.

20. An improvement in a browser adapted for use upon a client system to access a host, the browser adapted for use in connection with an admission control system that admits and defers messages on a priority basis, wherein the admission control system includes an admission control gateway and a deferral manager, and the deferral manager both sends to the client system a deferral message indicating that a previously submitted message has been deferred and includes with the deferral message an indication of a time when the deferred message can be later processed by the host, said improvement comprising:

a buffer adapted to store at least one host address and the indication of time;

a clock that determines a time when the deferred message can be processed by the host; and

means coupled to the clock for sending at the time determined by the clock a deferred message to the admission control system that enables admission of the deferred message on a priority basis.

21. An improvement according to claim 20, wherein the buffer is further adapted to store a key that must be presented to the admission control system for admission on a priority basis.

22. An improvement according to claim 20, wherein the means for sending a deferred message includes means for automatically opening a new browser window and for automatically sending the deferred message in association with the new browser window.

23. An improvement according to claim 20, wherein the means for sending a deferred message includes means for sending the deferred message in association with the new browser window at election of a user of the browser.

24. An improvement according to claim 20, wherein:

the buffer is adapted to store a plurality of host addresses and corresponding indications of time; and

means coupled to the clock for sending each deferred message to the admission control system for admission on a priority basis at proper time determined in response to the corresponding indication of time.

* * * * *

US005796719A

United States Patent [19]

Peris et al.

[11] Patent Number: 5,796,719

[45] Date of Patent: Aug. 18, 1998

[54] **TRAFFIC FLOW REGULATION TO GUARANTEE END-TO-END DELAY IN PACKET SWITCHED NETWORKS**

[75] Inventors: **Vinod Gerard John Peris**, Croton-on-Hudson, N.Y.; **Leonidas Georgiadis**, Thessaloniki, Greece; **Roch Andre Guerin**, Yorktown Heights, N.Y.; **Subir Varma**, San Jose, Calif.

[73] Assignee: **International Business Corporation**, Armonk, N.Y.

[21] Appl. No.: **659,458**

[22] Filed: **Jun. 6, 1996**

Related U.S. Application Data

[60] Provisional application No. 60/007,196, Nov. 1, 1995.

[51] **Int. Cl.**[6] **H04L 12/56**

[52] **U.S. Cl.** **370/231**; 370/230; 370/232; 370/412; 370/417; 370/418

[58] **Field of Search** 370/230, 231, 370/232, 233, 412, 417, 418

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,289,462	2/1994	Ahmadi et al.	370/60.1
5,347,511	9/1994	Gun	370/54
5,497,375	3/1996	Hluchyj et al.	370/94.1
5,625,622	4/1997	Johri	370/232
5,640,389	6/1997	Masaki et al.	370/418

OTHER PUBLICATIONS

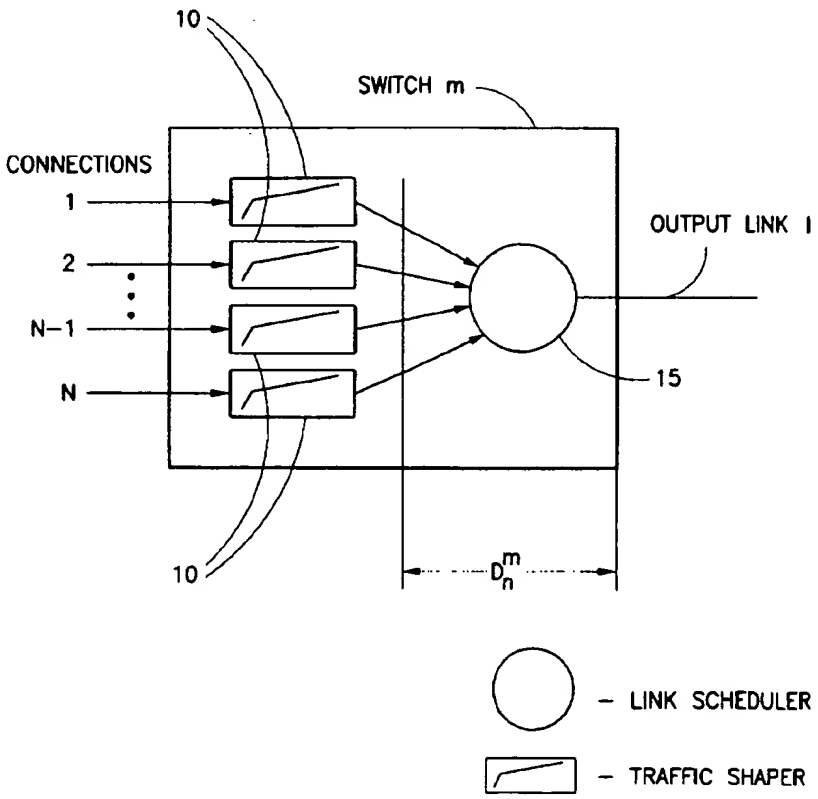
Hei Zhang and Domenico Ferrari, Rate-Controlled Service Disciplines, Journal of High Speed Networks 3(4),pp. 389–412, 1994.

Primary Examiner—Douglas W. Olms
Assistant Examiner—Wilbur T. Baker
Attorney, Agent, or Firm—Douglas W. Cameron

[57] **ABSTRACT**

The present invention relates to the issue of providing end-to-end delay guarantees in a multi-node communication system. More specifically, the present invention addresses the problem of specifying operational parameters of rate-controlled service disciplines in a communication network in order to efficiently provide end-to-end delay guarantees. The key contribution is a method for specifying leaky bucket parameters as well as scheduling delays at each node, which are used as inputs to the rate-controlled service discipline.

5 Claims, 3 Drawing Sheets



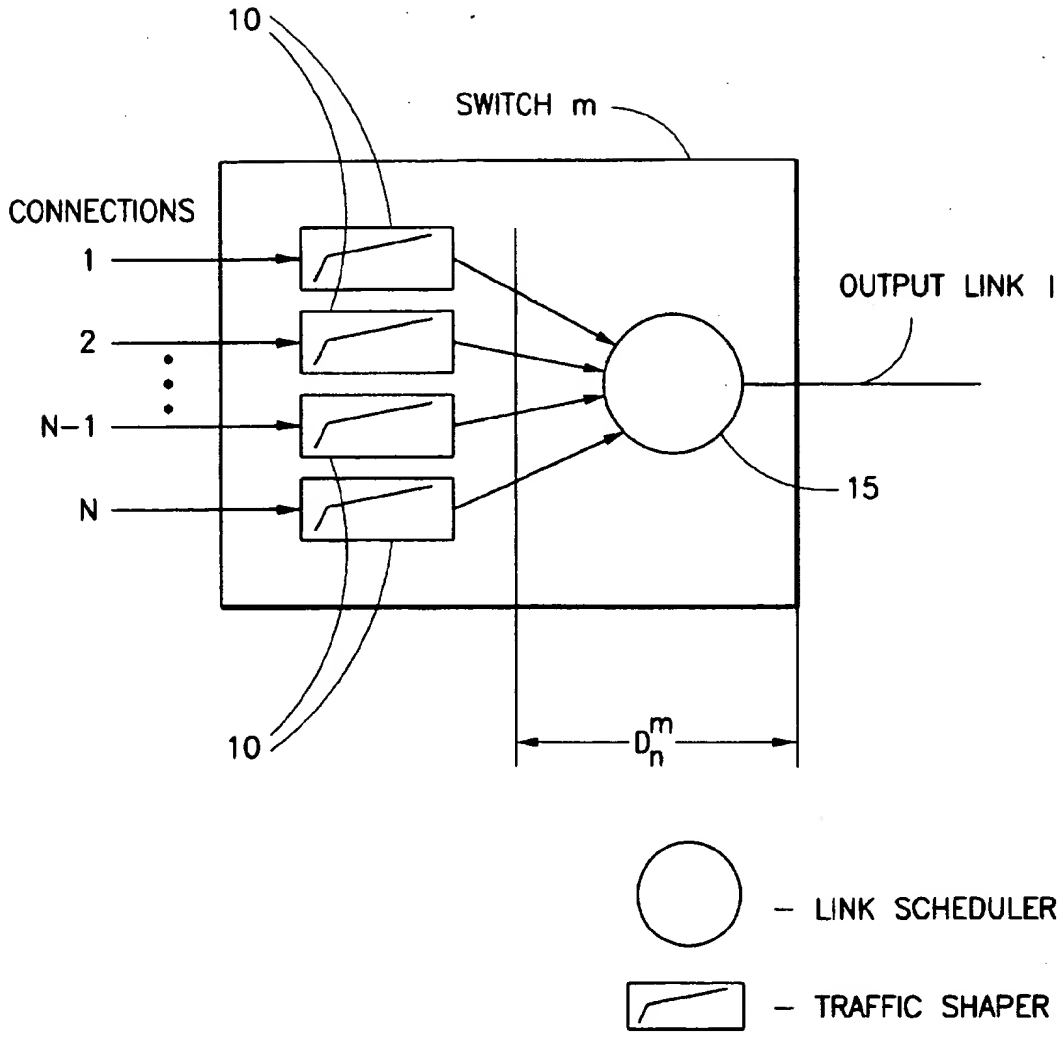

10
SWITCH m
CONNECTIONS
1
2
N-1
N
OUTPUT LINK I
15
10
D_n^m
– LINK SCHEDULER
– TRAFFIC SHAPER

FIG. 1

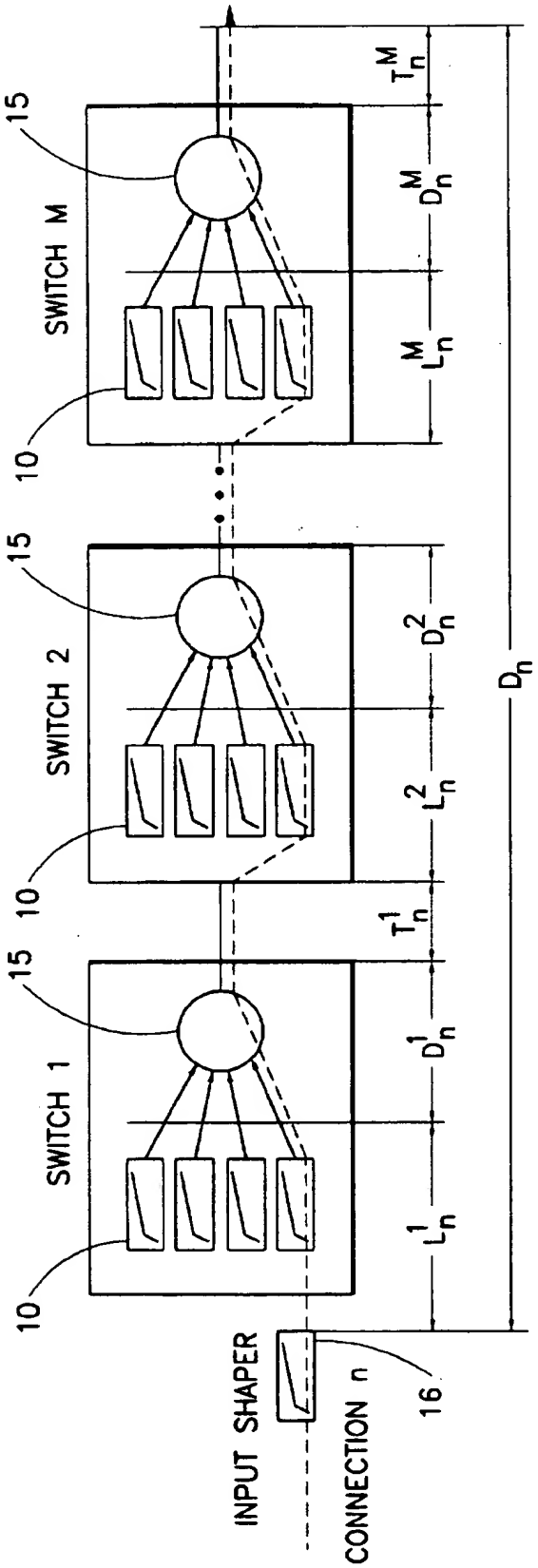

SWITCH M
SWITCH 2
SWITCH 1
INPUT SHAPER
CONNECTION n
15
10
16
T_n^M
D_n^M
L_n^M
D_n^2
L_n^2
T_n^1
D_n^1
L_n^1
D_n

FIG. 2

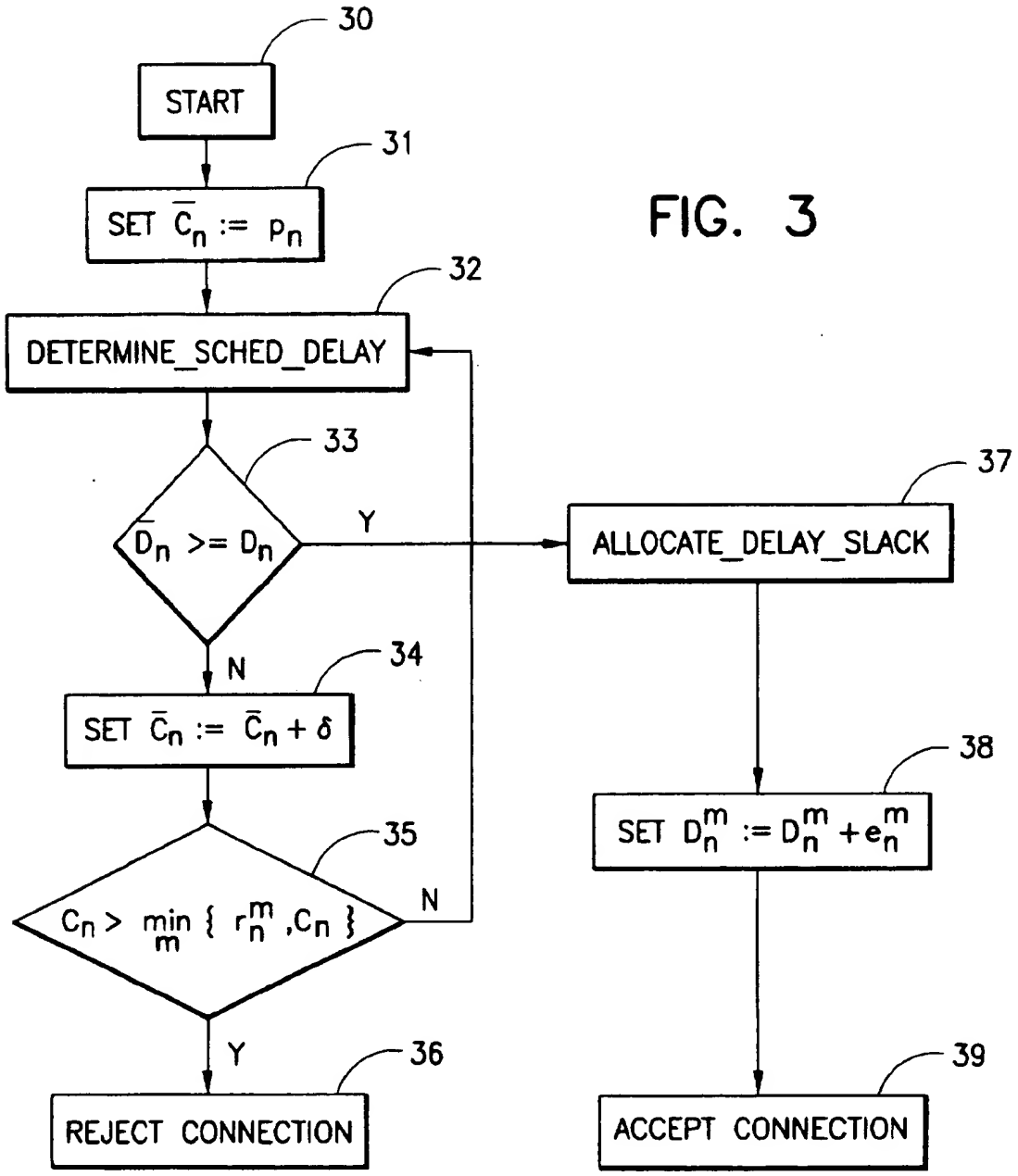

30
START
31
SET $\bar{C}_n := p_n$
32
DETERMINE_SCHED_DELAY
33
$\bar{D}_n >= D_n$
Y
N
34
SET $\bar{C}_n := \bar{C}_n + \delta$
35
$C_n > \min_m \{ r_n^m, C_n \}$
N
Y
36
REJECT CONNECTION
37
ALLOCATE_DELAY_SLACK
38
SET $D_n^m := D_n^m + e_n^m$
39
ACCEPT CONNECTION

FIG. 3

TRAFFIC FLOW REGULATION TO GUARANTEE END-TO-END DELAY IN PACKET SWITCHED NETWORKS

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application claims priority to co-pending U.S. provisional application 60/007,196, filed Nov. 1, 1995.

DESCRIPTION

1. Technical field

This invention describes a method and apparatus for regulating the flow of packets through a communication network to efficiently satisfy end-to-end delay guarantees. More specifically, this invention provides a mechanism for specifying the traffic shaper parameters and scheduler delay guarantees at each switch so that the connection's end-to-end delay requirements are satisfied.

2. Description of Prior Art

The problem addressed in this disclosure is that of designing per connection traffic shaping mechanisms at every switch of a packet-switched network in order to efficiently provide per packet end-to-end delay guarantees. Providing end-to-end delay guarantees is an important performance measure of many real-time applications such as video, voice, and remote sensing and process control. Various methods for providing end-to-end delay guarantees exist in the literature.

A frame structure is provided in [2]. Every packet arriving at a switch within one frame, is guaranteed to leave the switch at the next frame and thus the packet delay at that switch is guaranteed to remain below a certain bound that depends on the frame size. By having the frames at all switches synchronized, it is possible to provide overall delay guarantees as the sum of the delay guarantees at each switch. The number of connections that can be accommodated within one frame depends on the peak rate of these connections. Thus, in general, this method leads to inefficient use of the network since it requires that a connection be allocated bandwidth equal to its peak rate.

In [4], the employed service discipline at each switch is a non-preemptive policy that "tracks" the operation of the preemptive Generalized Processor Sharing policy (GPS) with general weights, by scheduling for transmission at the output link, the enqueued packet that would have completed transmission earliest under GPS. This packetized version of GPS is denoted by PGPS. Under the assumption that the input traffic to the network conforms to the output of a "leaky bucket" traffic shaper [5, 6], it is possible to analyze PGPS for fixed weights, and to provide end-to-end delay bounds that are in general much better than the bounds obtained by the simple addition of the worst case bounds at each switch. However, the method of obtaining these bounds is complicated since it requires to take into account the combined effect of all the connections at every switch through which a connection passes. Moreover, the connection admission protocol requires the solution of a inverse problem, namely that of determining the appropriate weights so that specified end-to-end delays guarantees are provided. This implies an even greater complexity for the connection admission protocol under the PGPS service discipline. However, for the specific PGPS policy where the assigned weights are proportional to the connection's maximum sustainable rate (or leaky bucket token generation rate), called Rate Proportional Processor Sharing (RPPS), simpler bounds can be obtained. Relative to the general PGPS policy, RPPS provides a greatly simplified method of obtaining end-to-end delay guarantees, but at the same time its flexibility in providing a wide range of these guarantees is significantly reduced. This is due to the fact that the simpler bounds obtained under RPPS are in general looser than the tight bounds of a POPS policy; moreover, since one has to pick a very specific weight assignment, the choice of possible delay guarantees that can be provided is limited.

A rate-controlled service discipline (RCSD) was first proposed in [7] and in general, it consists of two components. A leaky bucket traffic shaper, which regulates the the rate at which packets are made eligible for transmission on the output link, and a scheduler that arbitrates between the possibly many packets that are eligible for transmission, thus deciding the order of transmission of the eligible packets. A leaky bucket can be implemented by maintaining a counter which is periodically incremented until a certain threshold, that is larger than the maximum packet size. Whenever a packet arrives it is allowed to pass through the leaky bucket if the value of the counter is at least the size of the packet. Otherwise the packet is queued until the counter reaches the packet size. When a packet clears the leaky bucket, the counter is decremented by the size of the packet (See [3] for an example implementation). An example scheduler is the Non Preemptive Earliest Deadline First (NPEDF) scheduler. Here, each connection is associated with a certain delay guarantee. When the packets of a particular connection arrive at the scheduler, they are stamped with a deadline, which is the sum of their arrival time and the associated delay guarantee. Whenever the link is idle, the NPEDF scheduler picks the packet with the smallest deadline for transmission on the output link.

In the RCSD described in [7], the connection traffic at each switch passes first through a traffic shaper and is then delivered to the scheduler at the appropriate output link. The input traffic is assumed to conform to the output of a particular traffic shaper, hereafter referred to as "input traffic shaper". At each switch, the connection is first reshaped by a shaper identical to the input traffic shaper. It is shown that the end-to-end delay under this discipline can be obtained as the sum of the worst case delays at the scheduler at each switch, plus the propagation delay. That is, the shapers do not contribute additional delays to the end-to-end delay bounds. The major advantages of this class of service disciplines are simplicity of implementation, simplified analysis since each switch can be analyzed separately, and modularity since, depending on the processing power of the switch, scheduling policies of varying complexity and efficiency can be chosen. However, the end-to-end delay guarantees can be very loose since they are obtained as a sum of worst case delays at each switch.

A rate-controlled service discipline where the traffic shapers at each switch can reshape the connection's traffic to a shape that has, in general, different parameters than the input traffic shaper, is studied in [1]. It is shown that this modification has the potential to greatly improve the end-to-end delay guarantees and that it can also provide the tight delay guarantees of a PGPS policy. However, no general method for specifying the parameters of the traffic shapers and the scheduler delay guarantees is provided in [1].

References [1–9] are hereby incorporated herein by reference.

In this invention we consider a service discipline of the type described in [1] and provide a mechanism by which the traffic shaper parameters and the scheduler delay guarantees at each switch are specified for a connection during the

connection admission phase. In the special case where every connection requires the RPPS delay guarantees, the proposed mechanism can guarantee these delays as well. In addition, the proposed mechanism can provide a much wider range of delay guarantees.

SUMMARY OF THE INVENTION

Consider a packet-switched network which provides strict end-to-end delay guarantees by using a rate-controlled service discipline [7, 1]. The maximum packet length is L. It is assumed that the input traffic shaper of connection n (See 16 of FIG. 2) is of the type specified in the existing ATM [8] and proposed TCP/IP [9] standards. This is equivalent to assuming the following constraints on the connection traffic. Let $I_n[t,t+\tau]$ be the amount of connection n traffic that enters the network in the interval [t,t+T]. Then, for all t, $\tau \geq 0$,

$$I_n[t,t+\tau] \leq \tilde{I}_n(\tau) = L + \min\{c_n\tau, \sigma_n + \rho_n\tau\},$$

where c_n, σ_n, ρ_n, are referred to as the connection peak rate, burstiness and sustainable rate respectively. These parameters are negotiated with the network during the connection establishment process and are referred to as pre-determined traffic characteristics. These characteristics basically specify the input shaper envelope $\tilde{I}_n(\tau)$. Also, during the negotiation process, an upper bound on the end-to-end packet delay for connection n traffic, $\bar{D}_n$, is requested from the network. That is, $\bar{D}_n$ is the requested end-to-end delay guarantee.

The operation of a rate-controlled service discipline that will guarantee an upper bound $\bar{D}_n$ on the end-to-end packet delay for connection n traffic, requires the following specifications at each switch m (switching node m) along the path of the connection (See FIG. 1):

- The parameter values for a traffic shaper **10** (leaky bucket) L_n^m which reshapes the traffic of connection n at switch m, to conform to an envelope $L+\min\{c_n^m\tau, \sigma_n^m+\rho_n^m\tau\}$.
- Upper bounds on the scheduler delays at each of the output links of the switch. Given the leaky bucket parameters of all connections that are destined for output link l of switch m, the scheduling policy that arbitrates packet transmission on the link, guarantees an upper bound D_n^m for the scheduling delays (queueing in the scheduler **15** plus transmission time of a packet) of connection n packets.

It is known [1] that for any connection, it is optimal to have identical leaky bucket parameters at each switch along the path, but not necessarily identical to the parameters of the input traffic shaper. More specifically, we have that for each switch, m, along the path of the connection n,

$$\rho_n^m = \rho_n, c_n^m = \bar{c}_n, \sigma_n^m = \sigma_n \frac{1-\rho_n/\bar{c}_n}{1-\rho_n/c_n} = \bar{\sigma}_n.$$

It is also known [1] that the optimal value of $\bar{c}_n$ is in the range $[\rho_n, \min_m\{r_m^n, c_n\}]$, where r_m^n denotes the link capacity of the output link at node m to which the connection n traffic is routed. The previous specification still leaves the parameters $\bar{c}_n$ (peak leaky bucket rate) and D_n^m (scheduler delay) undefined. As shown in [1], these parameters are crucial for the efficient operation of the service discipline. The objective of this invention is to provide a method for determining $\bar{c}_n$ and D_n^m. First, however, we need to provide an expression for the end-to-end delay bound of a rate-controlled service discipline. Referring to FIG. 2, let connection n traverse switches 1, . . . , M, and let T_n^m be the propagation delay between switches m and m+1. By convention, switch M+1 (not shown in FIG. 2) is the destination of connection n. Let L_n^m be a bound on the delay the connection n traffic experiences in the traffic shaper of switch m (See FIG. 2). It can then be shown (See [1]), that the end-to-end delay, i.e., the delay a packet experiences from the time it exits the input traffic shaper (See 16 of FIG. 2) to the time it arrives at its destination, is bounded by

$$D_n = \begin{cases} L_n^1 + \sum_{m=1}^{M} (D_n^m + T_n^m), & \text{if } c_n > \rho_n \\ \sum_{m=1}^{M} (D_n^m + T_n^m), & \text{if } c_n = \rho_n \end{cases} \quad (1)$$

where

$$L_n^1 = \frac{\sigma_n}{c_n - \rho_n}\left(\frac{c_n}{\bar{c}_n} - 1\right),$$

Note that the bound on the delay includes only the shaper delays at the first switch (L_n^1). This shaper delay at the first switch is referred to as the "access delay bound". That is, the shaper delays, $L_n^2, L_n^3, \ldots, L_n^M$ are not included. The intuitive explanation for not including the shaper delays at the rest of the switches, is that only packets that were transmitted early by the scheduler at a switch, can be delayed by the shaper at the next switch. It is important to note that the upper bound on the scheduler delay at switch m, D_n^m, depends on the leaky bucket parameters and on the delay bounds of all the connections whose traffic is routed to the same output link at node m as the connection n traffic.

The method proposed in this invention determines $\bar{c}_n$, based on the following principle.

Make the peak rate $\bar{c}_n$ as small as possible as long as it can be guaranteed that $\bar{D}_n \geq D_n$.

The rationale for picking $\bar{c}_n$ as small as possible, is that this choice smoothes the traffic inside the network as much as possible and therefore will result in smaller buffer requirements at the switches.

Once the parameter $\bar{c}_n$ is determined, the upper bound on the delay at the scheduler, D_n^m, that switch m can provide to connection n has also been determined (See description of **32** of FIG. 3 below). The end-to-end delay D_n, can then be computed according to formula (1). It may, however, happen that we still have $\bar{D}_n > D_n$. In which case, it remains to determine how the delay slack, $\bar{D}_n - D_n$, is to be reallocated to the switches along the path of the connection. This reallocation is of importance as it allows us to relax the delay bounds of the connection at the switches, so that more future connection requests can be accepted. In this disclosure, we also describe a method to perform this reallocation. The method reflects the following objective

- Allocate the delay slack to the M switches that connection n traverses, so that the minimum delay bound over all the nodes is maximized.

The rationale for this objective is that it tends to equalize the allocated delay guarantees at all nodes, which in turn implies that the scheduler buffering requirements of the connection are spread more evenly among the switches.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is an illustration of the switching node m with the output link l on to which connections 1, . . . , N are multiplexed. Packets from each connection are first regulated by leaky buckets **10**, to ensure that the resulting output is conformant with the leaky bucket characteristics. Scheduler **15** controls the order in which packets that have cleared their respective leaky buckets are transmitted on the output link l.

FIG. 2 graphically illustrates the path of the connection through a number of switches or switching nodes. It also illustrates the access delay bound, L_n^1, as well as the scheduler delay bounds, D_n^m. The propagation delay along the link from switch 1 to switch 2 is indicated by T_n^1.

FIG. 3 is a flowchart illustrating the calculation of the peak rate and the upper bounds on the scheduler delays that can be used at each of the switching nodes.

DESCRIPTION OF PREFERRED EMBODIMENT

In this section we describe an implementation of the computations outlined in the previous section, to determine the appropriate peak leaky bucket rate, $\bar{c}_n$, of a new connection n with given end-to-end delay requirements, $\bar{D}_n$, as well as the upper bound on the scheduler delay, D_m^n, to be assigned at each node m on the connection's path. The inputs to the computations are the predetermined traffic characteristics of the connection $L_n=(c_n, \sigma_n, \rho_n)$, its desired end-to-end delay bound $\bar{D}_n$, and the path (set of nodes and links) over which the connection is to be routed. The implementation relies on the use of standard programming methods using high level programming languages such as C or C++ and a general purpose processor.

A flow-chart illustrating the different phases of the computations is shown in FIG. 3, which displays the iterative process used to determine the optimal value for $\bar{c}_n$. For the sake of clarity we describe the algorithms in this invention, in the framework of a centralized controller that controls each switch in the network. Such a controller can be implemented using any general purpose processor. Alternatively, the implementation may be distributed among the switches along the path of the connection with each switch only aware of the connections that are routed through it. Iterations of the process would then take place across the switches instead of at the centralized controller.

We first describe a simple component/subroutine that, as shown in FIG. 3, is used repeatedly in the determination of $\bar{c}_n$. Consider a link l at node m along the path through which connection n is routed. Assume that a value $\bar{c}_n$ has been tentatively selected for the peak rate of connection n, so that its characteristics at the output of the shaper at switch m (and at all other switches on the path) are now given by the triplet $L_n=(\bar{c}_n, \rho_n, \bar{\sigma}_n)$. Node m already has a set $C_{l,m}$ of connections established on link l, each with already determined characteristics $L_k^m=(\bar{c}_k^m, \rho_k^m, \bar{\sigma}_k^m)$ and associated delay bounds D_m^k, for all $k \in C_{l,m}$. The quantities L_k^m, D_k^m, $k \in C_{l,m}$ are stored in an array form at the centralized controller, or at node m in the case of a distributed implementation. Similarly, the propagation delay on the link from switch m to m+1, denoted by T_n^m is stored as part of the link attributes of the switch m, and is independent of the connection characteristics. All quantities are preferably represented using a floating point representation for greater precision, but a fixed-point representation can also be used if required.

Given tentative characteristics $(\bar{c}_n, \rho_n, \bar{\sigma}_n)$ for connection n and the knowledge of the quantities L_k^m, D_k^m, for all connections $k \in C_{l,m}$ already established on link l at node m, a key component in determining the final values for the network leaky bucket parameters, is to compute the minimum delay that can be guaranteed to connection n on link l of node m. This computation has to be carried out at each node m on the path, and is performed by the subroutine

$$\text{Determine\_Sched\_Delay}\ (\bar{c}_n, \bar{\sigma}_n, \rho_n, \{L_k^m, D_k^m, k \in C_{l,m}\})$$

(See 32.)

The implementation of this subroutine depends on the scheduling policy employed at the switch, and can be performed using standard operations available in any high level programming languages. An example of such a computation can be found in [1] for the case where the scheduler is an Earliest-Deadline-First (EDF) scheduler.

We are now ready to proceed with the description of the implementation, outlined in FIG. 3, of the algorithm used to determine the appropriate value of $\bar{c}_n$ as well as the upper bound on the scheduler delay, D_n^m at each node m on the path of connection n. The first step 31 of the algorithm is to select initial values for the network leaky buckets which in this case are determined by a single value $\bar{c}_n$ by setting

$$\bar{c}_n=\rho_n,\ \bar{\sigma}_n=\sigma_n\frac{(1-\rho_n/\bar{c}_n)}{(1-\rho_n/c_n)},$$

and determining if this results in a feasible end-to-end delay guarantee. The choice for such a starting point is that, as mentioned in the previous section, we are trying to minimize $\bar{c}_n$, and if the smallest possible value is feasible, the algorithm stops right after this first step.

In order to determine if $\bar{c}_n=\rho_n$ is feasible, we need to compute an upper bound on the scheduler delay, D_n^m, that can be guaranteed under this setting at each node m on the connection's path. The value D_n^m at each node m on the path of the connection n, is computed by applying the subroutine

$$\text{Determine\_Sched\_Delay}\ (\rho_n, 0, \rho_n, \{L_k^m, D_k^m, k \in C_{l,m}\}),$$

where we have set $\bar{c}_n:=\rho_n$ and $\bar{\sigma}_n:=0$. Once the scheduler delay bounds D_n^m have been computed for each node on the path, the next step is to determine the associated end-to-end delay bound D_n from equation (1). Recall, that in addition to the sum of the scheduler delay bounds, the end-to-end delay bound, D_n, includes the access delay bound (L_n^1), incurred at the shaper in the first switch, as well as the sum of the propagation delays (T_n^m) that are incurred at each of the links along the path. Two cases must then be distinguished depending on the value of D_n.

1. $\bar{D}_n \geq D_n$: (See 33.) In this case, the algorithm to determine $\bar{c}_n$ completes after this first step, and the required end-to-end delay bound can be achieved by setting $\bar{c}_n:=\rho_n$, i.e., after reshaping the traffic with a leaky bucket of type $(\rho_n, 0, \rho_n)$.

Since the desired peak rate of the connection is now known, the allocation of any residual "delay slack" $d_n=\bar{D}_n-D_n$ among the M nodes can now be performed. This amounts to increasing the scheduling delay D_n^M at node m by a quantity e_n^m, such that $\Sigma_{m=1}^M e_n^m=d_n$. This delay slack allocation is performed by the subroutine

$$\text{Allocate\_Delay\_Slack}\ (d_n, \{D_n^m, e_n^m, m=1, \ldots, M\}),$$

(See 37.)

which can be seen in FIG. 3 and will be detailed later. Note that this subroutine again relies on standard computational procedures which can be implemented using any high level programming language.

The last step is to set $D_n^m:=D_n^m+e_n^m$ (See 38.) and the connection is accepted 39.

2. $\bar{D}_n<D_n$: In this case the end-to-end delay bound cannot be met with $\bar{c}_n=\rho_n$, and it is necessary to consider larger peak rate values. The algorithm then searches for the smallest possible value larger than ρ_n such that the resulting end-to-end delay bound verifies $\bar{D}_n \geq D_n$.

This is achieved by means of a standard iterative process (the loop on the left-hand-side of FIG. 3), for which a pseudocode description is provided below. In each iteration the value of $\bar{c}_n$ is incremented by a fixed quantity δ, i.e., we

set $\bar{c}_n := \bar{c}_n + \delta$, where $\delta = (\min_m\{r_n^m, c_n\} - \rho_n)/\Delta$, and the parameter Δ determines the accuracy of the solution (See 34.). For each new value of $\bar{c}_n$, a check is made to test if the delay bound is met, i.e., if $\bar{D}_n \geq D_n$. The algorithm stops at the first peak rate value $\bar{c}_n$ for which this condition is true. As mentioned earlier, this is consistent with the principle of selecting the lowest possible peak rate. Once a feasible value has been identified for $\bar{c}_n$, the remaining delay slack is allocated to the different nodes so that the final local delay bounds D_n^m are determined. This computation (right-hand-side of FIG. 3) is again carried out based on the subroutine Allocate_Delay_Slack ( ), whose pseudocode implementation is also provided below. Finally, if the algorithm reaches $\bar{c}_n = \min_m\{r_n^m, c_n\}$ and we still have $D_n > \bar{D}_n$ (See 35.), the connection is rejected as it cannot be admitted in the network (See 36.), at least not with the requested delay guarantees.

Next we provide a pseudocode description of the above algorithm.

Algorithm for Determining $\bar{c}_n$ and Upper Bounds on Scheduler Delays

1. $\bar{c}_n := \rho_n$; $\delta := (\min_m\{r_n^m, c_n\} - \rho_n)/\Delta$
2. while ($\bar{c}_n \leq \min_m\{r_n^m, c_n\}$)
   (a) for ($m := 1$ to M)
      i. $D_n^m :=$ Determine_Sched_Delay ($\bar{c}_n$, $\bar{\sigma}_n$, ρ_n, $\{L_k^m, D_k^m, k \in C_{l,m}\}$)
   (b) if ($c_n > \rho_n$)
      i. $D_n := \sigma_n(c_n - \bar{c}_n)/((c_n - \rho_n)\bar{c}_n) + \sum_{m=1}^{M} (D_n^m + T_n^m)$
   (c) else $D_n := \sum_{m=1}^{M} (D_n^m + T_n^m)$
   (d) if ($\bar{D}_n \geq D_n$)
      i. $d_n := \bar{D}_n - D_n$
      ii. Allocate_Delay_Slack (d_n, $\{D_n^m, e_n^m, m = 1, \ldots, M\}$)
      iii. $D_n^m := D_n^m + e_n^m$
      iv. output $\bar{c}_n$, $\{D_n^m, m = 1, \ldots, M\}$; stop.
   (e) else $\bar{c}_n := \bar{c}_n + \delta$
3. reject connection; stop.

The above pseudocode implements a standard sequential search in the range of values that $\bar{c}_n$ can take. It may be possible to devise methods other than a sequential search that are more efficient. This would, however, require knowledge of the relation between D_n^m and $\bar{c}_n$, which in turn depends on the specific scheduling policy used. As was mentioned before, this is also the case for the subroutine Determine_Sched_Delay ( ) as the minimum possible delay obviously depends on the selected scheduling policy. The algorithm whose pseudocode is provided here applies irrespective of the policy chosen.

Step 2. (d) in the above pseudocode allows us to determine the local delay bounds at all the switches on the path of the connection, once the smallest possible value for $\bar{c}_n$ has been identified. The local delay bounds are obtained from the minimum delays D_n^m obtained for that value of $\bar{c}_n$, to which a fraction of the residual delay slack d_n is added. This requires that we specify the sub-routine

Allocate_Delay_Slack (d_n, $\{D_n^m, e_n^m, m=1, \ldots, M\}$).

As mentioned in the Summary of the Invention Section, the goal of this subroutine is to allocate the delay slack d_n to the M switches traversed by connection n, so that the minimum delay over all the nodes is maximized. In other words, the most stringent delay requirement is maximized.

The formulation of the corresponding optimization problem is as follows:

$$\max_{e_n^m} \left\{ \min_{1 \leq m \leq M} (D_n^m + e_n^m);\ e_n^m \geq 0,\ \sum_{m=1}^{M} e_n^m = d_n \right\}.$$

A solution to this problem is obtained as follows. First, an attempt is made to allocate the delay slack among the switches so as to yield equal delay bounds at all switches. This may, however, not be possible since it may actually require the lowering of some delay bounds, say at switch m, below the computed minimal value D_n^m. This actually means that these switches should not consume any of the available delay slack since their local bound is already higher than that at other switches. Switches that fall in this category are identified by the fact that they receive a negative delay slack allocation when trying to equalize all switch delays. When any such switch is present, the one with the largest D_n^m is removed from the computation, and another attempt is made at equalizing the delay bounds at the remaining switches. The process is repeated until a feasible allocation (no negative value) is achieved.

The subroutine Allocate_Delay_Slack ( ) implements the above procedure and its pseudocode representation is as follows.

subroutine Allocate_Delay_Slack(d_n, $\{D_n^m, e_n^m, m = 1, \ldots, M\}$)
1. Sort D_n^m; Let $D_n^{m1} \geq D_n^{m2} \geq \ldots \geq D_n^{mM}$.
2. for ($K = 1$ to M)
   (a) $c := \left(\sum_{i=K}^{M} D_n^{mi} + d_n \right)/(M - K + 1)$
   (b) if ($c \geq D_n^{mK}$)
      i. $e_n^{mi} := c - D_n^{mi}$, $K \leq i \leq M$, return
   (c) else
      i. $e_n^{mK} := 0$

The above subroutine can be easily modified to solve a slightly different optimization problem, where the equalization is performed not on the delay bounds, but instead only on their component due to queueing delays. In general, the delay bound of connection n at node m is of the form $D_n^m = L/r_n^m + q_n^m$, $q_n^m \geq 0$, where L/r_n^m represents the transmission delay of a maximum length (L) packet, while q_n^m corresponds to the scheduler queueing delay. It may be desirable to equalize as much as possible the buffering requirements due to queueing rather than total delays at the M switches that connection n traverses. This can be achieved through a simple modification of the above subroutine where we simply replace D_n^m with q_n^m, i.e., we have

Allocate_Delay_Slack (d_n, $\{q_n^m, e_n^m m=1, \ldots, M\}$).

Example of Application of the Algorithm to Achieve RPPS Delay Bounds

Last, we illustrate how the general algorithm described in this disclosure can be used to provide, as a special case, delay allocations and bounds equal to those of the RPPS policy. Assume now that the input traffic for connection n satisfies

$I_n(t,t+\tau] \geq \bar{I}(\tau) = L + \sigma_n + \rho_n \tau,$

and that the RPPS delay guarantees are required by all the connections in the network, i.e., for connection n it is required that

$$\bar{D}_n = \frac{\sigma_n + L + 2(M-1)L}{\rho_n} + \sum_{m=1}^{M} \left(\frac{L}{r_n^m} + T_n^m \right).$$

If the scheduling policy employed by each node is Non-preemptive Earliest Deadline First, (NPEDF), and we invoke the subroutine

Allocate_Delay_Slack (d_n, $\{q_n^m, e_n^m, m=1, \ldots, M\}$),

then using the techniques in [1] it can be shown that the resulting traffic reshaping parameters result in a rate-controlled service policy that guarantees the RPPS bounds.

REFERENCES

[1] L. Georgiadis, R. Guérin, V. Peris, and K. N. Sivarajan. Efficient Network QoS Provisioning Based on Per Node Traffic Shaping. Research Report RC 20064. IBM, T. J. Watson Research Center, May 1995.

[2] S. Jamal Golestani. A Framing Strategy for Congestion Management. *IEEE Journal of Selected Areas in Communication*, 9(7):1064–1077, September 1991.

[3] R. Guérin, A. K. Parekh, P. Chang and J. T. Rayfield. A Method and System for Implementing Multiple Leaky Bucket Checkers Using a Hybrid Synchronous/Asynchronous Update Mechanism. U.S. patent Ser. No. 08/382,464 filed Feb. 1, 1995, assigned to the same assignee of the current application. IBM Docket YO994–224.

[4] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.

[5] M. Sidi, W. Z. Liu, I. Cidon, and I. Gopal. Congestion Control Through Input Rate Regulation. In Proceedings of the *IEEE GLOBECOM* '89, pages 1764–1768, 1989.

[6] Jonathan S. Turner. New Directions in Communications (or which way to the Information Age?). *IEEE Communications Magazine*, 24(10):8–15. October 1986.

[7] Hui Zhang and Domenico Ferrari. Rate-Controlled Service Disciplines. *Journal of High Speed Networks*, 3(4) :389–412, 1994.

[8] ATM (Asynchronous Transfer Mode) *User Network Interface Specification, Version* 3.1, ATM Forum, September 1994.

[9] C. Partridge. A Proposed Flow Specification, Internet Network Working Group, Request for Comments: 1363, September 1992.

Having thus described our invention, what we claim as new and desire to secure by Letters Patents is:

1. In a communications network having a plurality of packet switching nodes and providing a requested end-to-end delay guarantee to a sequence of packets of a connection having pre-determined traffic characteristics, each of said nodes having a leaky bucket and scheduler to regulate the flow of said packets therethrough, said network thus having a plurality of leaky buckets and schedulers, a method of transmitting said packets of said connection through a path of said network, said method comprising:

(a) setting initial values for parameters of said leaky buckets of said nodes of said path based upon said pre-determined traffic characteristics of said connection;

(b) determining an access delay bound for said packets of said connection from said pre-determined traffic characteristics of said connection and said values for said parameters;

(c) determining upper bounds on the scheduler delays that would be experienced by said packets of said connection at those of said nodes of said path based upon said values for said parameters;

(d) updating said values for said parameters if the sum of said access delay bound and said upper bound on scheduler delays as well as the propagation delays for all of said nodes on said path is outside a fixed interval;

(e) repeating steps (b) through (d) with said updated values for said parameters determined in step (d), instead of with said initial values, until said sum is within said interval; and

(f) transmitting said packets of said connection through said path, where the flow of said packets will be regulated at each of said nodes by a corresponding one of said leaky buckets and a corresponding one of said schedulers, where said updated values for said parameters for said leaky buckets are set in accordance with said updated values for said parameters determined in step (e), so as to ensure that each of said packets of said connection will be transmitted through said path within said end-to-end delay guarantee.

2. A method as recited in claim 1, wherein said updates in step (d) are performed by incrementing said values for said parameters, and said fixed interval is from 0 to said end-to-end delay guarantee requested by the connection.

3. A method as recited in claim 1, further comprising:

increasing said upper bound on said scheduler delays at each of said nodes on said path, if said sum, determined from last repetition of step (e) is less than the end-to-end delay guarantee requested by the connection, so that said sum becomes equal to said requested end-to-end delay guarantee.

4. A method as recited in claim 1, wherein said initial values of said parameters are set so that the peak rate parameter of said network leaky buckets at each of said nodes on said path equals the sustainable rate of the said connection.

5. In a communications network having a plurality of packet switching nodes and providing a requested end-to-end delay guarantee to a sequence of packets of a connection having pre-determined traffic characteristics, each of said nodes having a leaky bucket and scheduler to regulate the flow of said packets therethrough, said network thus having a plurality of leaky buckets and schedulers, an apparatus for transmitting said packets of said connection through a path of said network, said apparatus comprising:

(a) means for setting initial values for parameters of said leaky buckets of said nodes of said path based upon said pre-determined traffic characteristics of said connection;

(b) means for determining an access delay bound for said packets of said connection from said pre-determined traffic characteristics of said connection and said values for said parameters;

(c) means for determining a bound on scheduler delays that would be experienced by said packets of said connection at those of said nodes of said path based upon said values for said parameters;

(d) means for updating said values for said parameters if the sum of said access delay bound and said upper bound on scheduler delays as well as propagation

delays for all of said nodes of said path is outside a fixed interval;

(e) means for repeating steps (b) through (d) with said updated values for said parameters determined in step (d), instead of with said initial values until said sum is within said interval; and

(f) means for transmitting said packets of said connection through said path, where the flow of latter said packets will be regulated at each of said nodes by a corresponding one of said leaky buckets and a corresponding one of said schedulers, where said updated values for said parameters for said leaky buckets are set in accordance with said updated values for said parameters determined in step (e), so as to ensure that each of said packets of said connection will be transmitted through said path within said end-to-end delay guarantee.

* * * * *